z14xcotrin.f

A set of subroutines to interpolate the OPAL opacity tables in the six input variables Z, X, exC, exO, T6, and R (where R = RHO / T6^3). Optionally, this can be extended by allowing interpolation in opacities for compositions where the C, N, O, and Ne in the metallicity Z have been interconverted by nuclear burning, and/or the OPAL opacity tables can be extended by molecular opacities (at low temperatures) and/or conductive opacities (at high densities).

Send any questions/comments/bug-reports to   boothroy@cita.utoronto.ca

TABLE OF CONTENTS:

```
                      ===============================
                      INTRODUCTION AND UPDATE HISTORY
                      ===============================

-----Arnold I. Boothroyd: VERSION of SEPTEMBER 7, 2007----------z14xcotrin21.f

  (Default version of this program needs 23.9 Mb of matrix storage.)

------------------------------------------------------------------------------
  OPAL 1995: Reference: C.A. Iglesias & F.J. Rogers (1996), ApJ, 464, 943-953
------------------------------------------------------------------------------
  Note OPAL opacities are available at http://www-phys.llnl.gov/Research/OPAL/
  This program is from  http://www.cita.utoronto.ca/~boothroy/kappa.html
******************************************************************************
  1995 OPAL opacities for arbitrary metallicity Z (in the range 0.0 to 0.1),
  including C,O-rich opacities.  Interpolation in Z can be performed when the
  opacities are read in and/or when the opacity-calculation routine is called;
  this choice is under user control.  It is possible to obtain opacities at
  arbitrary values of [O/Fe] by using the non-CO-rich opacity files 'GN93hz'
  and (one of) 'Alrd96a2', 'C95hz', or 'W95hz' (but this can only be done at
  the time when the opacities are being read in, not subseqently when the
  opacity-calculation routine is called).  Arbitrary hydrogen abundances and
  arbitrary amounts of excess carbon and oxygen are always allowed.  Uses the
  40 OPAL opacity files  Gz???.x??  (also checks for earlier names  Gx??z* ).
--ALSO: accurate opacity values can be interpolated as a function of varying
  relative C, N, O, and Ne abundances (and/or a user-specified element).
--ALSO: molecular opacities may be read in and used (at low T).
--ALSO: conductive opacities may be read in and used (at high density).
******************************************************************************
  Based on the old OPAL routines xcotrin.f and xcotrinz.f, modified to use
  the new (1995) opacity tables Gz???x?? and GN93hz --- similar to xcotrin21.f
  (but with a significantly different interface, as described further below).
******************************************************************************


------------------------------------------------------------------------------
*** UPDATED SEPTEMBER 7, 2007: from version of AUGUST 26, 2007:
 --Added automatic handling of the new alpha-enhanced GS98 molecular opacities
 that are now available from  http://webs.wichita.edu/physics/opacity/
------------------------------------------------------------------------------
*** UPDATED AUGUST 26, 2007: from version of APRIL 6, 2007:
 --Fixed a bug so that FZEDGE is now set correctly to 0.0 for Z out-of-range
 (previous version incorrectly set FZEDGE = 1.0 in such a case, and returned
 the conductive opacity [if available], rather than a value of 1.0E+35 to
 indicate that the opacity could not be computed; this bug is now fixed).
 --Smoother switch-off of non-exC,O-opacity-shifts for 0.2 < exC+exO < 0.3
 (this change only affects 0.2 < exC+exO < 0.3 at X > 0.03 [such compositions
 should seldom actually occur], and even there has almost negligible effect).
------------------------------------------------------------------------------
*** UPDATED APRIL 6, 2007: from version of MARCH 29, 2007:
 --Minor modifications, for smoother extrapolation in temperature and density
  (this will also improve the switchover from OPAL to conductive opacities).
 --Fixed a bug that caused incorrect CNO-interpolation at X = 0.03 (actually,
  for 0.029999 < X < 0.030001), yielding errors (usually, small ones) there.
 --Fixed a bug that could yield an incorrect name for the default [O/Fe] file.
------------------------------------------------------------------------------
*** UPDATED MARCH 29, 2007: from version of MARCH 24, 2007:
 --Added updated conductive opacities: from 'Potekhin et al.' (2006).
------------------------------------------------------------------------------
*** UPDATED MARCH 24, 2007: from version of FEBRUARY 2, 2007:
 --Simplified quadratic interpolation routines to reduce call overhead.
 --Added option to print out the names of the opacity files that were read in.
 --Fixed bug in Hubbard & Lampe conductive opacity interpolation (that had
  been introduced by quadratic-interpolation formula of FEBRUARY 2, 2007).
------------------------------------------------------------------------------
*** UPDATED FEBRUARY 2, 2007: from version of DECEMBER 16, 2006:
 --Added support for Ferguson et al. 2005 molecular opacities (essentially,
  updated versions of the Alexander and Ferguson 1994 molecular opacities).
 --Fixed bug in exC- and exO-interpolation at small helium abundances.
 --Modified quadratic-interpolation formulae to reduce roundoff error.
------------------------------------------------------------------------------
*** UPDATED DECEMBER 16, 2006: from version of NOVEMBER 23, 2006:
 --Allow bzip2 (de)compression of input files (suffix ’.bz2’) as well as gzip
  (suffix ’.gz’) and compress (suffix ’.Z’).
------------------------------------------------------------------------------
*** UPDATED NOVEMBER 23, 2006: from version of SEPTEMBER 21, 2006:
 --Some updates in the comments below, and a minor bug-fix.
------------------------------------------------------------------------------
*** UPDATED SEPTEMBER 21, 2006: from versions of SEPTEMBER 6 and 13, 2006:
 --Fixed the (allowed) extension of OPAL opacities using conductive opacities;
  such extension now works correctly, when specified by the user.
------------------------------------------------------------------------------
*** UPDATED SEPTEMBER 6, 2006: from previous version of OCTOBER 20, 2004:
 *****************************************************************************
 * NOTE that BINARY OPACITY DUMPS of previous versions are INCOMPATIBLE with *
 * this new version, due to changes in the arrays: any such binary dumps     *
 * MUST BE RECOMPUTED using  DUMP_OPAL_OPAC  with this new program version.  *
 *****************************************************************************
 --New, simpler subroutines have been added for reading in the opacities.
 --Alexander & Ferguson (1994) molecular opacity routines and storage added,
  with relevant subroutines to control its use (by default, it is NOT used).
 --Conductive opacity routines added, allowing use of Hubbard & Lampe (1969),
  & Itoh et al. (1983), Mitake et al. (1984) [by default these are NOT used].
 --Common block /opalGS98mixes/ revised, to hold the meteoritic Z-mix as well.
 --Minor bug fixed in computation of opacities that require the user-defined
  (nonCNO-interpolation) file CF_USER.
 --Added new routines:  ASK_OPAL_Z_MIX  to return abundances of components of
  the metallicity Z, and a few others involving opacity filename control.
------------------------------------------------------------------------------
*** UPDATED OCTOBER 20, 2004: from version of JUNE 30, 2004:
 --Common blocks revised, to reduce the file-size of the compiled program (by
  refraining from initializing any parts of very large common blocks).
------------------------------------------------------------------------------
*** UPDATED JUNE 30, 2004: from version of MARCH 9, 2004:
  --Some redundant variables removed, to avoid compiler warning messages.
------------------------------------------------------------------------------
*** UPDATED MARCH 9, 2004: from version of FEBRUARY 1, 2004:
  --Minor bug-fix in subroutine OPAL_X_CNO_FU, to handle more correctly the
  previously-ignored possibility that N+Ne may have decreased relative to Z
  (the bug-fix does its best to prevent exC and exO from becoming negative).
------------------------------------------------------------------------------
*** UPDATED FEBRUARY 1, 2004: from version of JANUARY 9, 2004:
  --Added subroutines  DUMP_OPAL_OPAC  and  READ_OPAL_DUMP  which allow one to
  store a set of opacities in unformatted binary form for future re-use (gives
  a large speed advantage when later one wishes to read opacities with the
  same inputs, at the cost of some disk space for the opacity dumpfile).
  --Added subroutines  ASK_KHIGHZ_OFE  and  ASK_MAIN_OPAL_FILE.
  --Moved all initializations of common block variables into a BLOCK DATA, as
  required by some compilers; fixed a few other minor bugs.
------------------------------------------------------------------------------
*** UPDATED JANUARY 9, 2004: from version of AUGUST 2, 2003: minor changes:
  --Added subroutines  ASK_LAST_OPAC  and  ASK_LAST_XCNOU  (which just return
  values contained in common blocks /E_OPAL_Z/ and /X_OPAL_Z/, respectively).
  --Fixed subroutine OPAL_X_CNO_FU so that it no longer yields an error if one
  calls this subroutine OPAL_X_CNO_FU before calling the subroutine READZEXCO.
  --Fixed the places where some compilers objected to the use of the string
  concatenation operator.
-------------------------------------------------------------------------------
*** UPDATED AUGUST 2, 2003: from version of APRL 27, 2001: Add the option of
  using an alternate set of OPAL files (e.g., 'GS98hz' rather than 'GN93hz')
  to get opacities appropriate to an updated solar composition (e.g., Grevesse
  and Sauval 1998) --- note that the files  Gz???.x??  need not be updated, as
  their opacities are shifted to agree with those from the file  'GS98hz' .
  ALSO, add the option of interpolating the opacities as a function of varying
  relative C, N, O, and Ne abundances (as well as in the "excess" C and O).
  ALSO, increase the allowed file-plus-directory name length to 255 characters
  (rather than 80); this affects  common /opdir/  and the alternate OPAL file
  set, but not  common /opalmixes/  (default OPAL files, 8 characters long).
  Also, maximum allowed T6,R extrapolation is now just over one grid spacing
  (instead of just under); this latter change should have negligible effect.
  Also, an approximation used previously when computing the abundances for
  mixes that are interpolated in [O/Fe] has been replaced by the exact formula
  (this change should also have little effect: none at all for [O/Fe]=0.0).
-------------------------------------------------------------------------------
*** UPDATED APRIL 27, 2001: from version of MARCH 4, 2001: for more accurate
  X-interpolation at X > 0.1 (with a LARGE improvement in the accuracy as X
  approaches 1-Z), using the added X-values available in the file  GN93hz  .
  Also fixed a minor bug in the CO-interpolation (that could have caused small
  errors in the interpolated opacity, but only in the seldom-encountered
  situation of near-maximal CO-enrichment combined with a non-zero hydrogen
  abundance: X > 0.0 with C+O just less than 1-X-Z).
-------------------------------------------------------------------------------
*** UPDATED MARCH 4, 2001: from version of MAY 28, 1999: add the metallicity Z
  to the list { X, C, O, T6, R } of variables in which the OPAL opacities can
  be interpolated (rather than being restricted to only a single metallicity,
  defined when the opacities were read in).  This is required if gravitational
  settling of metals takes place.  Also, newly added subroutines allow easier
  opacity-directory specification, and easier user control of how Z, T, and R
  edges and extrapolation are handled.  Also, the opacity files  Gz???x??  and
  GN93hz  are allowed to be in compressed form (they must have suffix  .gz  if
  they were compressed using 'gzip', or suffix  .Z  if they were compressed
  using 'compress'); any compressed opacity files will be decompressed, read
  in, and compressed again.
-------------------------------------------------------------------------------
  Updated MAY 28, 1999: from version of JUNE 26, 1997: to look for the CO-rich
  opacity file names in the newer format  Gz???.x??  before trying the old
  format  Gx??z*  .  Note that some opacity values in  Gz001.x35 Gz004.x70
  Gz050.x35 Gz100.x70  differ by roundoff amounts (+/- 0.001 in log10[kappa])
  from corresponding older files  Gx35z001 Gx70z004 Gx35z05 Gx70z10  ; also,
  Gz050.x35 and  Gz100.x70  each omit a redundant duplicate composition table
  present in  Gx35z05  and   Gx70z10  .  In all other cases, the newer OPAL
  files  Gz???.x??  are identical to the corresponding older files  Gx??z* .
******* ALSO: the name of the common block that returns the opacity values
  has been changed from  common/e/  to  common/e_opal_z/  in order to avoid
  compilation errors when compiling using  f2c  (Fortran to C conversion).
-------------------------------------------------------------------------------

*******************************************************************************


        TEMPERATURES AND DENSITIES CONTAINED IN OPAL OPACITY TABLES:
        *************************************************************


  The OPAL opacities are tabulated in terms of logT and logR, where
    logR = logRHO - 3 * ( logT - 6 )  , i.e.,  R = RHO / T6^3
  (with T being temperature in Kelvins and RHO being density in g/cm^3)
  The "density" and temperature ranges contained in the tables are:
    logR : -8.0 to 1.0   [at logR = -8.0 (0.5) 1.0, i.e., delta_logRHO = 0.5]
    logT : 3.75 to 8.70  [at logT = 3.75 (0.05) 6.00 (0.10) 8.10 (0.20) 8.70]
```

```
          COMPOSITIONS FOR WHICH OPACITY TABLES ARE AVAILABLE:
          ****************************************************

Type 2 OPAL Tables - including enhanced C & O (40 files):
=========================================================
These are 40 files of the form  Gz???.x??  , where the "z???" part may be
any of { "z000", "z001", "z004", "z010", "z020", "z030", "z050", "z100" }
and the "x??" part may be any of { "x00", "x03", "x10", "x35", "x70" }.
These have 8 metallicities Z = { 0.0, .001, .004, .01, .02, .03, .05, .1 }
and 5 hydrogen abundances X = { 0.0, 0.03, 0.1, 0.35, 0.7 }; each pair of
{ X, Z } has up to 60 different compositions with varying amounts of
"excess" carbon and oxygen (above that contained in Z), i.e., mixes having
exC,exO = { 0.0, .01, .03, .1, .2, .4, .6, 1-X-Z } (such that no mix has
X+Z+exC+exO > 1.0).  These files allow fairly good interpolated opacities
for X < 0.75 and Z < 0.12, with any amount of excess carbon and oxygen.
This may suffice if there is not much diffusion (to yield high X values),
but these opacities are NOT AT ALL ACCURATE for X > 0.75 (very high hydrogen
abundances); opacities at X = 1-Z may be off by up to an order of magnitude
(unless corrected by including the Type 1 tables below).

The above files are ALWAYS read in, for one or more Z-values (as determined
by your call to one of the opacity-reading subroutines).

Type 1 OPAL Tables - fixed metal distribution (by default, 4 files):
====================================================================
file 'GN93hz'   [O/Fe]=0.0: solar composition, Grevesse and Noels (1993)
file 'Alrd96a2' [O/Fe]=0.3: alpha enhanced elements, Allard (1996)
file 'C95hz'    [O/Fe]=0.4: alpha enhanced elements, Chaboyer (1995)
file 'W95hz'    [O/Fe]=0.5: alpha enhanced elements, Weiss (1995)
Each of these 4 files contains 126 compositions: opacities at 13 Z-values
Z = {0.0,0.0001,0.0003,0.001,0.002,0.004,0.01,0.02,0.03,0.04,0.06,0.08,0.1}
and at 10 X-values X = {0.0,0.1,0.2,0.35,0.5,0.7,0.8,0.9,0.95,1-Z}; they do
NOT have enhanced-CO ("excess-C,O") compositions.

One or two of these Type 1 files will typically be read in (as determined by
your call to one of the opacity-reading subroutines).  For non-CO-rich cases
this allows slightly improved Z-interpolation (for Z < 0.12) and slightly
improved X-interpolation (for 0.03 < X < 0.75); for high hydrogen abundances
(X > 0.75), such as may result from diffusion (e.g., helium settling), the
accuracy is GREATLY IMPROVED.

NOTE THAT additional Type 1 tables can be computed at the OPAL website, or
may be available elsewhere.  By default, this program alows for counterparts
of the above 4 files with a different composition (e.g., for the Grevesse &
Sauval 1998 mix: 'GS98hz', 'GS98hz_OFe.3_Alrd96a2', 'GS98hz_OFe.4_C95', and
'GS98hz_OFe.5_W95', or the more recent Asplund, Grevesse, & Sauval 2004 mix:
'AGS04hz', etc.).  This program also allows for the existence of files where
C, N, O, and Ne abundances are interconverted (as by nuclear burning: e.g.,
'GN93hz.CtoN', 'GN93hz.COtoN', 'GN93hz.CNOtoNe'); such files can be used
to enable the program to return accurate opacity values as a function of
varying relative C, N, O, and Ne abundances.  Some other user-specified
element (or set of elements) can also be interpolated (e.g., 'GN93hz.user').

The program can also be set to use  'AGS04hz'  etc., instead of the above.


Additional (OPTIONAL) non-OPAL tables: conductive and molecular opacities:
************************************* ===================================

-- They are available for the sake of convenience (to allow extension of the
     OPAL opacities to lower T and higher R), and some testing has been done
     to check that the switchover between tables works reasonably, but the
     USER MUST CHOOSE whether to read in (and use) these extensions or not.
```

```
  'Potekhin et al.' (2006) CONDUCTIVE OPACITIES: opacity file 'condall06.d'
  =========================================================================
  -- Conductive opacities available:  http://www.ioffe.rssi.ru/astro/conduct/
******************************************************************************
*****  NOTE: as of March 9, 2007: the revised/updated file  condall06.d  *****
*****  containing conductive opacities is available on the website.     *****
*****  This should be an improvement over the Hubbard & Lampe (1969)    *****
*****  plus Itoh (1984) conductive opacities described further below.   *****
******************************************************************************
  These 'condall06.d' electron conductivities are available in the ranges:
    -6 < logRHO < 9.75    in density RHO, with spacing  delta logRHO = 0.25
    3 < logT < 9          in temperature T, with spacing  delta logT = 1/3
    Zion = {1,2,3,4,6,8,12,16,20,26,30,36,42,48,60}    mean nuclear charge

-----FROM THE ABOVE WEBPAGE:  "The plasma is assumed fully ionized (electron
  collisions with neutrals are neglected, the ions are assumed pointlike).
  This model may be still useful for evaluation of conductivities of partially
  ionized plasmas, if one uses a mean-ion model.  Then the ion charge Zion
  should be replaced by an effective (or average) ion charge Zeff.  For
  nondegenerate plasmas, the results are based on a continuation from the
  degenerate domain (using Fermi-Dirac averaging) and can be considered as
  order-of-magnitude estimates.  For degenerate plasmas, on the contrary, the
  results come from the exact theory and should be much more accurate."
    "Non-magnetic conductivities have been updated 18 July 2006.  The updated
  codes and table have been put at this site 9 September 2007.  There are two
  modifications.  1. The electron-electron (ee) scattering contribution is
  now included in such a way that both the cases of strongly degenerate and
  nondegenerate plasmas are recovered accurately.  Now the high-T limit of our
  data matches the numerical tables of Hubbard & Lampe, 1968, ApJS 18, 297
  (which remain the most accurate calculations of conductive opacities for
  astrophysical use in nonmagnetic, nondegenerate, weakly coupled plasma).
  2. The ee contribution is updated according to the results by Shternin and
  Yakovlev (2006)."

-----References for the above (non-magnetic) electron thermal conductivities:
    A. Y. Potekhin, D. A. Baiko, P. Haensel, D. G. Yakovlev (1999), "Transport
  properties of degenerate electrons in neutron star envelopes and white
  dwarf cores", Astron. Astrophys. 346, 345
    O. Y. Gnedin, D. G. Yakovlev, A. Y. Potekhin (2001), "Thermal relaxation
  of young neutron stars", Mon. Not. R. Astron. Soc. 324, 725
    P. S. Shternin, D. G. Yakovlev (2006), "Electron thermal conductivity
  owing to collisions between degenerate electrons", Phys. Rev. D 74, 043004
    S. Cassisi, A. Y. Potekhin, A. Pietrinferni, M. Catelan, M. Salaris (2007)
  "Updated electron-conduction opacities: the impact on low-mass stellar
  models", Astrophys. J., in press [astro-ph/0703011]

-----NOTE: the last reference above suggests that the rms ionic charge is the
  appropriate one for use in plasmas not so dense as to be crystalline, i.e.,
    Zion = (<z^2>)^0.5 = sqrt( SUM_i{ z_i * z_i * n_i } / SUM_i{ n_i } )
         = sqrt( SUM_i{ z_i * z_i * X_i / A_i } / SUM_i{ X_i / A_i } )
  where z_i is the ionic charge of element i and n_i is its number density;
  n_i is proportional to X_i / A_i, the mass fraction over the atomic mass.
    [Correctly summing the contributions of different ions in a multicomponent
  plasma would be "much more complicated", according to the references.]
    Also: the log of the electron thermal conductivity  logCHI  (stored in the
  file condall06.d) converts to the log of the conductive opacity  logKAPPA_c
  via:
       logKAPPA_c = log( 16 * sigma / 3 ) + 3 * logT - logRHO - logCHI
  where
       sigma = the Stefan-Boltzmann constant.
  - By default, if conductive opacities are read in, then the (OPAL) opacity-
  calculating subroutines will actually return the overall effective opacity
    Kappa = 1 / ( 1 / Kappa_radiative + 1 / Kappa_conductive )  .
  This will be allowed to extend the opacities to the highest densities for
  which valid conductive opacities are available (significantly higher than
  the upper density limit of the OPAL opacity table, at most temperatures).


  The above are preferable to the older available conductive opacities:

  W.B. Hubbard & M. Lampe (1969) ApJS, 163, 297: H&L opacity file 'Condopac'
  ==========================================================================
  -- In a form usable by these opacity subroutines, this file is available
       only from   http://www.cita.utoronto.ca/~boothroy/kappa.html
       (the same website as for this file containing the opacity subroutines).
  This file includes 'H&L' conductive opacities for H, He, and C, for somewhat
  irregular temperature ranges in the density range -6 < logRHO < 6.  When
  this file is read in, approximate tables for O and Ne are also stored (by
  adding a constant to the table for C).

  'Itoh' (1984) conductive opacities (to supplement the above input file)
  =======================================================================
  - By default, the 'H&L' tables are superseded by the later 'Itoh' conductive
  opacities at high density (where the latter are valid), using formulae from
  N. Itoh, S. Mitake, H. Iyetomi, & S. Ichimaru (1983), ApJ 273, 774, plus
  S. Mitake, S. Ichimaru, & N. Itoh (1984), ApJ 277, 375.  [The formulae are
  used to get the combined effects of conductive opacities for H, He, C, O,
  and Ne, where Ne represents all elements heavier than oxygen.]

  - "Reasonable" interpolated (and extrapolated) conductive opacity values are
  computed in regions where neither of the above are valid.


  J.W. Ferguson et al. (2005) ApJ, 623, 585: (updated) MOLECULAR OPACITY files
  ============================================================================
  -- Opacity files available from  http://webs.wichita.edu/physics/opacity/
*****************************************************************************
*****  NOTE that these 2005 molecular opacities should supersede and    *****
*****  replace those of Alexander & Ferguson (1994) below; the authors  *****
*****  state that the accuracy should be significantly better.          *****
*****************************************************************************
  The Ferguson et al. (2005) molecular opacity table ranges are:
    2.7 < logT < 4.5  in temperature T  [one table may be missing logT < 2.8]
                        at logT = 4.5 (-0.05) 3.5 (-0.01) 2.9 (-0.05) 2.7
    -8.0 < logR < 1.0  in R = RHO / T6^3  [this is same range as OPAL tables]
                        at logR = -8.0 (0.5) 1.0
    Z = { 0.0 0.00001 0.00003 0.0001 0.0003 0.001 0.002 0.004 0.01 0.02 0.03
            0.04 0.05 0.06 0.08 0.1 }
    X = { 0.0 0.1 0.2 0.35 0.5 0.7 0.8 0.9 0.95 1-Z }
  Each of the 155 (X,Z) cases has its own ASCII file of opacity values, with
  the file name reflecting the values of X and Z (as per examples below).
  Several different mixes are available, each with its set of 155 (X,Z) cases
  (contained in a gzip-compressed tar archive on the above web page); for most
  of these, compositions are also specified:
  --- GN93 mix (Grevesse & Noels 1993): tar archive  f05.g93.tar.gz  contains
   g0.0.tron g0.00001.tron ... g99999.00001.tron g10.0.tron
  --- GS98 mix (Grevesse & Sauval 1998):  f05.gs98.tar.gz  contains
   g98.0.0.tron g98.0.00001.tron ... g98.99999.00001.tron g98.10.0.tron
  --- L03 mix (Lodders 2003):  f05.l03.tar.gz  contains
   l03.0.0.tron l03.0.00001.tron ... l03.99999.00001.tron l03.10.0.tron
  --- AGS04 mix (Asplund, Grevesse & Sauval 2004):  f05.ags04.tar.gz  contains
   ags04.0.0.tron ags04.0.00001.tron ... ags04.99999.00001.tron ags04.one.tron
  Compositions are listed on the website for these mixes.  There are also two
  mixes for which opacities are available but the composition is not listed:
  --- S92 mix (Seaton 1992) -- composition unspecified, but mix is presumably
  very similar to GN93 (opacities are very similar):  f05.s92.tar.gz  contains
   s92.0.0.tron s92.0.00001.tron ... s92.99999.00001.tron s92.one.tron
  --- S92AE mix (alpha-element-enhanced version of S92) -- composition not
  specified, except for [alpha/Fe] = [O/Fe] = 0.3:  f05.s92ae.tar.gz  contains
   s92ae.0.0.tron s92ae.0.00001.tron ... s92ae.99999.00001.tron s92ae.one.tron
  --- alpha-enhanced mixes for Grevesse & Sauval 1998 (the compositions for
  these are listed on the website):
    --- GS98-.2 with [alpha/Fe] = [O/Fe] = -0.2: f05.gs98-.2.tar.gz contains
         gs98-.2.0.0.tron ... gs98-.2.99999.00001.tron gs98-.2.one.tron
    --- GS98+.2 with [alpha/Fe] = [O/Fe] = 0.2: f05.gs98+.2.tar.gz contains
         gs98+.2.0.0.tron ... gs98+.2.99999.00001.tron gs98+.2.one.tron
    --- GS98+.4 with [alpha/Fe] = [O/Fe] = 0.4: f05.gs98+.4.tar.gz contains
         gs98+.4.0.0.tron ... gs98+.4.99999.00001.tron gs98+.4.one.tron
    --- GS98+.6 with [alpha/Fe] = [O/Fe] = 0.6: f05.gs98+.6.tar.gz contains
         gs98+.6.0.0.tron ... gs98+.6.99999.00001.tron gs98+.6.one.tron
    --- GS98+.8 with [alpha/Fe] = [O/Fe] = 0.8: f05.gs98+.8.tar.gz contains
         gs98+.8.0.0.tron ... gs98+.8.99999.00001.tron gs98+.8.one.tron

***Any excess C or O is simply added to Z, yielding LESS ACCURATE OPACITIES.

  - Opacities are considered to be O.K. for Ztotal < 0.10, and Ztotal > 0.12
  is considered to be "unavailable" (where Ztotal = Z + exC + exO).

  - By default, the switchover from OPAL to Ferguson et al. 2005 opacities
  will occur smoothly as logT is reduced from 4.4 to 4.2 (this is the
  temperature range in the overlap region where the two opacity tables seem
  to be the closest to being equal).


  D.R. Alexander & J.W. Ferguson (1994) ApJ, 437, 879: opacity file 'Alexopac'
  ============================================================================
  -- In a form usable by these opacity subroutines, this file is available
       only from   http://www.cita.utoronto.ca/~boothroy/kappa.html
       (the same website as for this file containing the opacity subroutines).
******************************************************************************
*****  NOTE that these 1994 molecular opacities are SUPERSEDED by the    *****
*****  2005 opacities from   http://webs.wichita.edu/physics/opacity/    *****
*****  (see Ferguson et al. 2005, ApJ, 623, 585), as DESCRIBED ABOVE.    *****
******************************************************************************
  This file includes molecular opacities and dust opacity; table ranges are:
    3.0 < logT < 4.1  in temperature T,
    -12.0 < logRHO < -6.0  in density RHO (not R, unlike OPAL), for
    Z = { 0.0001, 0.0003, 0.001, 0.002, 0.004, 0.01, 0.02, 0.03, 1.0 }  and
    X = { 0.0001, 0.03, 0.10, 0.35, 0.70, 0.80 }  .
  [note that the Z = 1.0 table is of course only available for X = 0.0].
  Extensions of this table:
  - The X = 0.0001 tables are treated as having X = 0.0 (the difference is
  negligible for the above metallicity range 0.0001 < Z < 0.03).
  - Unlike the OPAL 'GN93hz' opacities, these must be extrapolated for X > 0.8
  but this yields errors no larger than the difference between Alexander and
  OPAL opacities even up to X = 1 - Z.
  - To match the available OPAL tables, tables at Z = 0.05 and Z = 0.10 were
  extrapolated from the Z = 0.02 and 0.03 cases, taking the average of the
  result of extrapolating Kappa in Z and of logKappa in logZ (this yielded the
  best results in the temperature region of overlap between Alxander and OPAL
  opacities).
  - A set of { Z = 0.0, X = 0.7 } opacities have been obtained using Fig. 1 of
  Alexander & Ferguson 1994, and the OPAL opacities at logT = 3.75 or higher.
  - A rough approximation of opacities for { Z = 0.0, X = 0.0 } was obtained
  by using the OPAL opacities at high-T, and extrapolating logKappa downwards
  in logT by fitting a least-squares quadratic.
  - For other X-values at Z=0.0 these two opacity sets were interpolated in X.

***This extends the tables of low-temparature opacities from 0.0001 < Z < 0.03
  to the full available metallitiocy range 0.0 < Z < 0.1 of the OPAL tables;
  but THE ACCURACY OF THE EXTENSIONS MAY BE POOR (especially for Z < 0.0001);
  it is better to use the Ferguson et al. (2005) molecular opacities instead.

***Also, any excess carbon or oxygen is simply added to Z: for significant
  "excess-C,O" (exC,exO comparable to Z) this yields LESS ACCURATE OPACITIES,
  and for large "excess-C,O" (0.1 < Z + exC + exO < 1.0) one must interpolate
  between the Z = 0.1 and Z = 1.0 mixes, yielding VERY INACCURATE OPACITES.

  - By default, if 'Alexopac' is read in, then the switchover from OPAL to
  Alexander opacities will occur smoothly as logT is reduced from 3.97 to 3.87
  (this is the temperature range in the overlap region where the two opacity
  tables seem to be the closest to being equal).

  - By default, Alexander opacities are considered to be O.K. for Z < 0.10 at
  all X, and for all Z at X = 0.0; they are considered to be "unavailable" in
  the region where Z > 0.15 at X > 0.03 (this constraint be tightened by the
  user at the time when the opacities are read in).


-------------------------------------------------------------------------------



                 ==========================================
                 List of subroutines contained in this file
                 ==========================================

  This file contains the following subroutines; the nine subroutines marked
  with  **  are those the user is most likely to wish to use (and are the
  first subroutines described below), while those marked with  *  can be
  called by the user (and are described in the comments further below):

    BLOCK DATA OPAL_OPAC_DATA
    SUBROUTINE OPALINIT( KHIGHZ, OFEBRACK, Z, KZ, KMET )
    SUBROUTINE GET_ZAVAIL
    SUBROUTINE GET_TRVALS
 ** SUBROUTINE OPAC( Z, XH, EXC, EXO, T6, R )
 ** SUBROUTINE OPAL( Z, XH, EXC, EXO, SLT, SLR )
 ** SUBROUTINE OPAL_X_CNO_FU( XH, SLT, SLR, XMET, NMET, FU )
  * SUBROUTINE OPAL_F_CNOU( Z, XH, EXC, EXO, SLT, SLR, FCN, FCON, FCNONE, FU )
    SUBROUTINE OPAL_F_XCON_CNOU( Z, XH, EXC, EXO, Y, XCN, XON, XNEHEAVY,
                   FMUAINV, FMUEINV, ZSQBAR, SLT, SLR, FCN, FCON, FCNONE, FU )
    SUBROUTINE Z_FCNO( XH,XMET,NMET,FU, Z,EXC,EXO, Y,XCN,XON,XNEHEAVY,
                       FMUAINV,FMUEINV,ZSQBAR, FCN,FCON,FCNONE,FUSE )
  * SUBROUTINE KAPFERG(SLT,SLR,XH,Z,EXC,EXO,FLKA,DLKATR,DLKARO,DLKAT,FKAEDGE)
  * SUBROUTINE CACHEFERG( KSTO, XH, XZCO )
  * SUBROUTINE KAP_MOL(SLT,SLR,XH,Z,EXC,EXO,FLKA,DLKATR,DLKARO,DLKAT,FKAEDGE)
  * SUBROUTINE KAPALEX( FLT, FLRO, X, XZCO, FLKA, DLKAT, DLKARO, FKAEDGE )
  * SUBROUTINE ASK_LAST_ALEX_EDGE( FKAEDGE, FTRA, FTRA_LO, FTRA_HI, FZKAEDGE )
  * SUBROUTINE KAPCOND( FLRO, FLT, X, Y, XCN, XON, XNEHEAVY, FMUAINV, FMUEINV,
                        ZSQBAR, IDER, FLKC, FLKCT, FLKCRO, FKCEDGE, FKCOK )
  * SUBROUTINE KAP_COND_POT( FLRO, FLT, ZION, IDER,
                             FLKC, FLKCT, FLKCRO, FKCEDGE, FKCOK )
  * SUBROUTINE OPAL_K_ONLY( Z, XH, EXC, EXO, SLT, SLR, FCN, FCON, FCNONE, FU )
 ** SUBROUTINE ASK_LAST_OPAC( OP, DOPT, DOPR, DOPTD, FEDGE, FTREDGE, FZEDGE )
  * SUBROUTINE ASK_LAST_XCNOU( Z, X, XC, XO, SLT, SLR, FCN, FCON, FCNONE, FU )
  * SUBROUTINE ASK_OPAL_Z_MIX( IMIX, XIZ, N_X, FNINZ, N_N )
  * SUBROUTINE ASK_OPAL_MIX_WT( ATWT, NWT, ATWTHHE, NHHE, ATZ, NZ )
 ** SUBROUTINE SET_OPAL_DIR( CDIRIN )
 ** SUBROUTINE SET_MOL_DIR( CDIR_MOL )
 ** SUBROUTINE SET_COND_DIR( CDIR_COND )
 ** SUBROUTINE SET_OPAL_LIST_LEVEL( LIST_LEVEL )
  * SUBROUTINE SET_OPAL_LIST_UNIT( LIST_IU )
 ** SUBROUTINE READ_BASIC_OPAL_OPAC( IU, Z, CF_HZ, OFEBRACK, CF_OFE )
 ** SUBROUTINE READ_EXTENDED_OPAC( IU, Z, CF_HZ, OFEBRACK, CF_OFE,
                                   I_MOL, I_COND, I_CNO, CF_USER )
  * SUBROUTINE SET_OFE_FILE( CFILEOFE )
  * SUBROUTINE SET_ALTMIX_OFE_FILE( CFILEOFE )
  * SUBROUTINE SET_METEOR_MIX_FILE( CFILEMET )
  * SUBROUTINE SET_ALTMIX_MAIN_FILE( CFILE_HZ )
  * SUBROUTINE SET_CNO_FILES( CF_HZ, CF_C, CF_O, CF_N, CF_U )
  * SUBROUTINE SET_CNO_EXT( IE, CE_HZ, CE_C, CE_O, CE_N, CE_U )
  * SUBROUTINE SET_COND_FILE( CFILECOND, I_FULL_PATH )
  * SUBROUTINE SET_COND_USE( KCOND, KREPLACE_ITOH )
  * SUBROUTINE SET_COND_INFLAGS( KCOND_FIX, KCOND_GAP, KCOND_HAVE )
  * SUBROUTINE ASK_COND_USE( KCOND, KCOND_AVAIL, KREPLACE_ITOH )
  * SUBROUTINE ASK_COND_INFLAGS( KC_FIX, KC_GAP, KC_FIX_N, KC_GAP_N )
  * SUBROUTINE SET_FERG_USER( CBEG_FERG )
  * SUBROUTINE ASK_FERG_USER( CBEG_FERG )
  * SUBROUTINE SET_FERG_ACC( IACC )
  * SUBROUTINE ASK_FERG_ACC( IACC )
  * SUBROUTINE SET_ALEX_FILE( CFILEALEX, I_FULL_PATH )
  * SUBROUTINE SET_ALEX_USE( KALEX )
  * SUBROUTINE SET_ALEX_DO_RHOSW( IRHOSW )
  * SUBROUTINE ASK_ALEX_USE( KALEX, KALEX_AVAIL, ITYPE )
  * SUBROUTINE ASK_KHIGHZ_OFE( KHIGHZ_USED, OFEBRACK_USED )
  * SUBROUTINE ASK_OPAL_FILE_USED( ITYPE, CF_USED )
  * SUBROUTINE SET_XHI( KXHI )
  * SUBROUTINE ASK_XHI( KXHI, KAVAIL )
  * SUBROUTINE SET_CNO_INTERP( KCNO, KUSER )
  * SUBROUTINE ASK_CNO_INTERP( KCNO, KUSER, KCNO_AVAIL, KUSER_AVAIL )
  * SUBROUTINE SET_ERR_CHECK( LEVEL )
  * SUBROUTINE ASK_ERR_CHECK( LEVEL )
  * SUBROUTINE SET_LOGT6_LIMITS( VLO, DVLO, VHI, DVHI )
  * SUBROUTINE SET_LOGR_LIMITS( VLO, DVLO, VHI, DVHI )
  * SUBROUTINE RESET_Z_LIMITS( VLO, DVLO, VHI, DVHI )
  * SUBROUTINE ASK_LOGT6_LIMITS( VLO, DVLO, VHI, DVHI )
  * SUBROUTINE ASK_LOGR_LIMITS( VLO, DVLO, VHI, DVHI )
  * SUBROUTINE ASK_Z_LIMITS( NZMAX, ZMIN, ZMAX )
  * SUBROUTINE ASK_Z_USE( NZUSE, ZLO, ZMID, ZHI, ZLOEX, ZHIEX )
  * SUBROUTINE ASK_Z_ARRAY( KZSTART, KARRAYSTART, ZARRAY, NARRAY )
  * SUBROUTINE SET_SMOOTH( INITSMOOTH, LOWCOSMOOTH, INTERPCOSMOOTH )
  * SUBROUTINE ASK_SMOOTH( INITSMOOTH, LOWCOSMOOTH, INTERPCOSMOOTH )
  * SUBROUTINE SET_LOGT_SW_FERG( FLTSW_LO, FLTSW_HI )
  * SUBROUTINE ASK_LOGT_SW_FERG( FLTSW_LO, FLTSW_HI )
  * SUBROUTINE SET_LOGT_SW_ALEX( FLTSW_LO, FLTSW_HI )
  * SUBROUTINE SET_LOGRHO_SW_ALEX( FLRHOSW_LO, FLRHOSW_HI )
  * SUBROUTINE SET_LOGT_RHOSW_ALEX( FLTSW_R_LO, FLTSW_R_HI )
  * SUBROUTINE ASK_LOGT_SW_ALEX( FLTSW_LO, FLTSW_HI )
  * SUBROUTINE ASK_LOGRHO_SW_ALEX( FLRHOSW_LO, FLRHOSW_HI )
  * SUBROUTINE ASK_LOGT_RHOSW_ALEX( FLTSW_R_LO, FLTSW_R_HI )
  * SUBROUTINE READCO( Z, KALLRD, KHIGHZ, IULOW )
  * SUBROUTINE READEXCO( Z, KALLRD, KHIGHZ, IULOW, OFEBRACK )
  * SUBROUTINE READZEXCO( NZIN, ZLO, Z, ZHI, KHIGHZ, IULOW, OFEBRACK )
  * SUBROUTINE READ_BEST_MOL( IU, I_MOL, CF_HZ, OFEBRACK )
  * SUBROUTINE READFERG( IU )
  * SUBROUTINE ADDFILE_FERG( IU, F_STO, F_READ, CFILEFERG, I_FULL_PATH )
  * SUBROUTINE READ_ADD_FERG( IU, F_STO, F_READ, KTYPE_FERG )
    SUBROUTINE FIND_FERG( KTYPE_FERG, LEN_BEG, CFILE_USE )
    SUBROUTINE INIT_FERG_INDICES
  * SUBROUTINE READALEX( IU )
  * SUBROUTINE READ_COND_POT( IU )
  * SUBROUTINE READCOND( IU )
```

```
   SUBROUTINE FIND_RJUMP_HL_COND
** SUBROUTINE DUMP_OPAL_OPAC( IU, CF_D )
** SUBROUTINE READ_OPAL_DUMP( IU, CF_D )
   SUBROUTINE READ_KZ( KZ, Z, KALLRD, KHIGHZ, IULOW, OFEBRACK )
   SUBROUTINE REVISE_HITR_FOR_INITSMOOTH
   SUBROUTINE COINTSMO( XXC, XXO, KZ )
   SUBROUTINE COINTERP( XXC, XXO, KZ )
   SUBROUTINE T6RINTERP( SLR, SLT )
   SUBROUTINE SNGL_CINTERP3(ZM,Z0,Z1,ZP,Z,N0,MXNV,VM,V0,V1,VP,VF,DF,D2,XH)
   SUBROUTINE QZLOG4INT( ZLOGD )
   FUNCTION QUADSL( IC, I, X, Y1, Y2, Y3, X1, X2, X3 )
   FUNCTION QDERSL( IC, I, X, Y1, Y2, Y3, X1, X2, X3 )
   FUNCTION QCHKSL( IC, I, X, Y1, Y2, Y3, X1, X2, X3 )
   FUNCTION QUAD( IC, I, X, Y1, Y2, Y3, X1, X2, X3 )
   FUNCTION QDER( IC, I, X, Y1, Y2, Y3, X1, X2, X3 )
   FUNCTION QCHK( IC, I, X, Y1, Y2, Y3, X1, X2, X3 )
   FUNCTION QZINTER( IC, I, Z, NMOREZ, F1, F2, F3, F4, Z1, Z2, Z3, Z4, ZDEL )
   FUNCTION MIXFIND( IU, IOFE, IGETZXI, IREW, ITAB, LINE, Z, X, C, O )
   SUBROUTINE CHK_DIR_NAME( CDIRIN, COPDIR, KOPE )
   SUBROUTINE CHK_OFE_ALT_FILE( K_OFE )
   SUBROUTINE INDEX_CO_DELTAS( ISET, KXHZ, JX, JC, JO )
   SUBROUTINE FINISH_CNO
   SUBROUTINE SPLINE( X, Y, N, Y2 )
   SUBROUTINE SPLINT( XA, YA, N, Y2A, X, Y, YP )
   SUBROUTINE FITY
   SUBROUTINE FITX
   SUBROUTINE GETD( F, N, D, FP1, FPN )
   SUBROUTINE INTERP( FLT, FLRHO, G, DGDT, DGDRHO, IERR )
   SUBROUTINE SMOOTH
   SUBROUTINE OPALTAB
   SUBROUTINE OPEN_CHK_ZIP( IU, FNAME, IGZIP, CMSG )
   SUBROUTINE CLOSE_CHK_ZIP( IU, FNAME, IGZIP )
   SUBROUTINE QUADSLSTO( I, X, X1, X2, X3 )
   FUNCTION QUADSLGET( I, Y1, Y2, Y3 )
   SUBROUTINE QDERSLSTO( I, X, X1, X2, X3 )
   SUBROUTINE QDERSLGET( I, Y1, Y2, Y3, Y, DYDX )
   SUBROUTINE QCHKSLSTO( I, X, X1, X2, X3 )
   FUNCTION QCHKSLGET( I, Y1, Y2, Y3 )
   SUBROUTINE QCHKSTO( I, X, X1, X2, X3 )
   FUNCTION QCHKGET( I, Y1, Y2, Y3 )
   SUBROUTINE QDERNSTO( I, NMORE, X, X1, X2, X3, X4 )
   SUBROUTINE QDERNGET( I, NMORE, Y1, Y2, Y3, Y4, Y, DYDX )
   SUBROUTINE QUADNSTO( I, NMORE, X, X1, X2, X3, X4 )
   FUNCTION QUADNGET( I, NMORE, Y1, Y2, Y3, Y4 )
   SUBROUTINE QDER4STO( I, X, X1, X2, X3, X3 )
   SUBROUTINE QDER4GET( I, Y1, Y2, Y3, Y4, Y, DYDX )
   SUBROUTINE QUAD4STO( I, X, X1, X2, X3, X3 )
   FUNCTION QUAD4GET( I, Y1, Y2, Y3, Y4 )
   SUBROUTINE QDERSTO( I, X, X1, X2, X3 )
   SUBROUTINE QDERGET( I, Y1, Y2, Y3, Y, DYDX )
   SUBROUTINE QUADSTO( I, X, X1, X2, X3 )
   FUNCTION QUADGET( I, Y1, Y2, Y3 )
   FUNCTION NUM_BLANKS_CONTAINED( FNAME )
   FUNCTION NON_BLANK_BEGIN( FNAME )
   SUBROUTINE OPOLDR( IU, FNAME )
   SUBROUTINE OPOLUF( IU, FNAME )
   SUBROUTINE OPNEUF( IU, FNAME )
   SUBROUTINE INQFIL( FNAME, LXST )
   SUBROUTINE LINUX_GET_HOME_DIR( FNAME, FNALT, IALT )
   FUNCTION LNBLNK( FNAME )
```

The last 6 of the above subroutines contain file-handling routines; if one is using VMS rather than some flavor of Unix or Linux, then one may have to comment out some statements in these subroutines and uncomment others, as well as in the "data cb" statement at the end of BLOCK DATA OPAL_OPAC_DATA. (The last 2 routines should be needed only if you are using fort77 under Linux, but should still work correctly on any flavor of Unix/Linux system.)

Note that the above routines have been tested on several Linux systems (and Unix, for some earlier versions), but have NOT been tested on a VMS system.

```
  **********************************************************
  NOTE THAT ALL REAL VARIABLES ARE SINGLE PRECISION (real*4)
  **********************************************************

  **********************************************************
  NOTE THAT NO FILENAME MAY EXCEED 255 CHARACTERS IN LENGTH.
  **********************************************************


------------------------------------------------------------------------------



         ==========================================================
         The subroutines that interpolate among the OPAL opacities:
         ==========================================================
```

-----NOTE that the following four opacity-calculating subroutines CAN extend the OPAL opacities to lower temperatures (by using molecular opacities) or to higher densities (using conductive opacities); whether they do either (or both) of these extensions DEPENDS ON HOW YOU READ IN THE OPACITIES via the subroutines described further below.

```
*** OPAC( z, xh, exC, exO, T6, R )     The purpose of the subroutines OPAC or
    ------------------------------     OPAL is to perform up to 6-variable
*** OPAL( z, xh, exC, exO, slt, slr )  interpolation on log10(kappa), to yield
    ---------------------------------  the opacity (and also its temperature
  and density derivatives) at the given composition, temperature, and density
  values (the details of how this interpolation is performed are discussed
  further below).  The user can control how the opacities are initially read
  in via subroutines discussed further below; otherwise, the first time OPAC
  or OPAL is called, opacities will be read in for an estimated "optimum"
  range of Z-values (that encompass the input value z).  These subroutines
  actually call OPAL_F_CNOU( z, xh, exC, exO, slt, slr, 0.0, 0.0, 0.0, 0.0 )
  (see description below) to perform the opacity interpolation.
```

The SINGLE-PRECISION REAL interpolation variables are:

```
       z      The metallicity, Z (excluding any "excess" C and O)
       xh     The hydrogen mass fraction, X
       exC    The enhanced ("excess") carbon mass fraction, exC.
                The total carbon mass fraction, Xc, is the sum of exC and
                the initial amount included in the metal mass fraction Z
       exO    The enhanced ("excess") oxygen mass fraction, exO.
 OPAC:
       T6     The temperature in millions of degrees Kelvin, T6
       R      = { rho(gm/cc) / T6**3 }, the temperature-shifted density value
 OPAL:
       slt    log10(T6) = log10(T) - 6
       slr    log10(R) = log10(rho) - 3 * slt = log10(rho) - 3 * [log10(T)-6]
```

(by definition, the helium mass fraction is Y = 1.0 - z - xh - exC - exO). Note that, while z and xh must be non-negative, small NEGATIVE values for exC and/or exO are allowed, provided that the sums { z + exC , z + exO , z + exC + exO } are non-negative; this leads to (linear) extrapolation in log(z+exC+0.001) and/or log(z+exO+0.001), unlike the earlier version of MAY 28, 1999 (where negative exC or exO values were treated as being zero).

Your routine that calls to OPAC or OPAL should either include the statement:

```
    common/e_opal_z/ opact,dopact,dopacr,dopactd,fedge,ftredge,fzedge
```

OR ELSE, after calling the opacity-calculation routine (e.g., OPAC or OPAL):

```
       call ASK_LAST_OPAC(OPACT,DOPACT,DOPACR,DOPACTD,FEDGE,FTREDGE,FZEDGE)
            ---------------------------------------------------------------
```

(this subroutine ASK_LAST_OPAC just returns the values from the common block /e_opal_z/ in its user-supplied arguments).

These SINGLE-PRECISION REAL variables have the following meanings:

```
    OPACT       returns the Log of the Rosseland mean opacity: Log10(kappa)
    DOPACT      returns dLog(kappa)/dLog(T6) at constant R (NOT rho!)
    DOPACR      returns dLog(kappa)/dLog(R) at constant T6, which is
                         = dLog(kappa)/dLog(rho) at constant temperature
    DOPACTD     returns dLog(kappa)/dLog(T6) at constant density, which is
                         = dLog(kappa)/dLog(T) at constant density
                         = DOPACT - 3.0 * DOPACR
    FEDGE       returns the degree-of-extrapolation product FTREDGE * FZEDGE
                      or, in some cases, A SMALLER VALUE (even zero):
                   - If you have read in conductive opacities, then FEDGE is
                      reduced in switchover regions (radiative to conductive
                      or one conductive table to another) where one or both
                      of the opacities requires some extrapolation (FTREDGE,
                      described below, is NOT reduced, to indicate that the
                      opacities should still be quite accurate there).
                   - If the 'GN93hz' opacities are NOT available, then FEDGE
                      reduces to zero as X increases from 0.76 to 0.8 (but
                      the opacity is still calculated out to X = 1 - Z); you
                      should ALWAYS SUPPLY the file 'GN93hz', since errors
                      can be quite large in some cases if you do not (up to
                      an order of magnitude for Z < 0.001 at X = 1 - Z).
    FTREDGE     returns 1.0 for T6,R inside table boundaries, reduces to 0.0
                      as T6,R moves more than one grid spacing outside table
                      (except, in general, in switchover regions).
    FZEDGE      returns 1.0 for Z inside the available range [zlow,zhigh],
                      reduces to 0.0 as Z moves out to the boundaries of the
                      extreme-Z-extrapolation range [zlo_ex,zhi_ex].
```

```
BY DEFAULT, the OPAL-opacity calculating routines set OPACT = 1.0E+35 and
 return without actually calculating the opacity IF:
   (1) FZEDGE = 0.0  , or
   (2) FTREDGE = 0.0 and ( logR < -6 or logT < 3.97 )  .
 Otherwise, even if FTREDGE = 0.0 , the radiative opacity is extrapolated
 (linearly) without limit to high T or R, since an estimate of the radiative
 opacity may be needed in such regions to determine whether the conductive
 opacity suffices there or not.
```

```
IF YOU HAVE READ IN CONDUCTIVE OPACITIES, then FTREDGE will be non-zero
 where they are valid and dominate, even if the radiative opacities are not
 valid there: i.e., conductive opacities can be used to EXTEND the radiative
 opacities, as well as being combined with them via the relevant formula
   Kappa = 1 / ( 1 / Kappa_rad + 1 / Kappa_cond )  .
Also: FTREDGE will be unity in the density region where opacities switch
 over from fully-valid radiative to fully-valid conductive ones, even if
```

```
   part of this switchover region requires some extrapolation of one or both
   of these.  However, FEDGE will be less than unity (possibly even zero) in
   switchover regions where such such extrapolation is required, and also in
   density gaps where different conductive opacity tables do not quite meet.
  (Also, at high T and R, an extrapolated conductive opacity value is returned
   even in regions where FTREDGE is zero.)

  NOTE THAT, if you have set the error-checking level to 2 (see SET_ERR_CHECK
   below), then ANY "out-of-range" case (where the final value of FEDGE = 0.0)
   is considered a fatal error and the program halts.

  If FZEDGE = 0.0, then the given Z-value lay too far ouside the available
   Z-range to be extrapolated (this is checked first, before T6 and R).
  If FTREDGE = 0.0 (and FZEDGE > 0.0), then the given T6 and/or R values lay
   too far outside the available table for reasonable extrapolation.

  Details of the extrapolation, and of subroutines allowing user control over
  the boundaries, are discussed further below.


*** OPAL_X_CNO_FU( xh, slt, slr, xmet, nmet, fu )  This subroutine adds any
    --------------------------------------------  opacity shifts due to the
  interconversions C --> N, O --> N, and/or N --> Ne (which can arise from
  nuclear burning) to the opacities interpolated in the 6 basic variables
  { z, xh, exC, exO, slt, slr }.  USE OF THIS SUBROUTINE CAN BE TRICKY.

 ---WARNING--- This subroutine estimates Z from the mass fraction of elements
  heavier than Ne.  UNLESS (1) you keep track of the mass fractions at least
  of C, N, O, Ne, and "heavies", and (2) your initial Z-composition, namely
  {C, N, O, Ne, "heavies"}, is THE SAME as that in the "solar" opacity table
  (e.g., 'GN93hz', 'GS98hz', or 'AGS04hz'), this subroutine will obtain an
  ERRONEOUS Z value and thus an INCORRECT OPACITY.

 ---WARNING--- If nmet = 19 in your array, then you must have initialized ALL
  of {C,N,O,Ne,Na,Mg,Al,Si,P,S,Cl,Ar,K,Ca,Ti,Cr,Mn,Fe,Ni} to a "solar" opacity
  table mix (e.g., from 'GN93hz' or 'GS98hz'), OTHERWSE you will get ERRONEOUS
  Z and CNO-interpolation factors (since the program uses stored atomic weight
  values to convert xmet into number fractions in this case), and thus an
  INCORRECT OPACITY.  If you include either fewer or more heavy elements,
  e.g., if you combine some of these elements into a collective "Xheavy", then
  you MUST use a value of nmet NOT EQUAL TO 19.

 ---WARNING--- OPAL_X_CNO_FU implicitly assumes that elements heavier than Ne
  are negligibly affected by nuclear burning.  IF ANY ELEMENTS HEAVIER THAN Ne
  ARE PRODUCED VIA NUCLEAR BURNING, this will cause the Z-value estimated by
  OPAL_X_CNO_FU to increase by roughly 5 to 6 times as much as the increase in
  heavy element abundance.  This may give LESS GOOD OPACITY values, or even
  yield estimated Z-values so large as to be OUT OF RANGE (this will occur for
  Xheavy > 0.02, roughly; in the worst case, Xheavy > 0.015 may give estimated
  Z > 0.1, i.e., beginning to be out of range).  If any elements heavier than
  Ne are produced via nuclear burning, you may wish to assign all or most of
  the newly-nucleosynthesized "heavies" to the Ne abundance, for purposes of
  opacity calculation (or else use some other subroutine).

  The input variables are:

       xh      The hydrogen mass fraction, X  (as for OPAL or OPAC above)
       slt     log10(T6) = log10(T) - 6  (as for OPAL above)
       slr     log10(R) = log10(rho) - 3 * slt  (as for OPAL above)
       xmet    SINGLE-PRECISION REAL ARRAY of size nmet, giving the mass
                 fractions of the "metals", i.e., of C, N, O, Ne, ...
                 NOTE that these are the actual mass fractions (NOT the mass
                 fractions relative to Z), and any "excess" C or O should be
```

```
                  included in the values of xmet(1) or xmet(3), respectively.
                  By definition, the mass fraction Y of helium is given by
                  Y = 1 − xh − SUM{xmet(i)}  , where i=1,...,nmet in the SUM.
       nmet     INTEGER size of the array xmet: ideally it should be the case
                  that  nmet = nel_zmix = 19  , in which case xmet is assumed
                  to hold the mass fractions of the elements of the OPAL mix,
                  namely, {C,N,O,Ne,Na,Mg,Al,Si,P,S,Cl,Ar,K,Ca,Ti,Cr,Mn,Fe,Ni}.
                  If the array size  nmet  is not 19, then the sum of xmet(5)
                  through xmet(nmet) is used as the total mass fraction of all
                  elements heavier than Ne, i.e., the array xmet must contain
                  at least {C,N,O,Ne,Xheavy}.
                NOTE that if Xheavy contains any part of the abundances from C,
                  N, O, or Ne, then the calculated Z value will be in error!
                (Also, if you have set the error level to 3 or higher [see the
                  subroutine SET_ERR_CHECK] and CNO−interpolation is available,
                  then it is a fatal error if nmet is not equal to 19.)
       fu       SINGLE−PRECISION REAL variable, giving the fraction of the
                  opacity shifts to be applied from any user−specified file: fu
                  multiplies opacity differences between the files CF_USER and
                  CF_HZ (as specified by the subroutine READ_EXTENDED_OPAC:
                  see below).
                If NO user−specified opacity file CF_USER has been read in,
                  then the value of  fu  is ignored (a value of 0.0 is used).

   This subroutine uses the array  xmet(nmet)  to calculate the metallicity Z,
   the excess carbon and oxygen EXC and EXO, and the fractions FCN, FCON, and
   FCNONE (to apply the C −−> N, O −−> N, and N −−> Ne opacity shifts −−− the
   CNO−interpolation of logKappa is linear in the CNO number fractions).  In
   general, it sets  FUSE = fu  (except that the value of FUSE is restricted
   so that it does not correspond to a reduction by more than a factor of 2 in
   the total number density of elements heavier than Ne).  This subroutine then
   does the equivalent of calling
     OPAL_F_CNOU( Z, xh, EXC, EXO, slt, slr, FCN, FCON, FCNONE, FUSE )  .

   NOTE that there would usually be little point in using OPAL_X_CNO_FU unless
   you have called   READ_EXTENDED_OPAC  with a non−zero value of  I_CNO  (to
   allow CNO−interpolation in the opacities) and/or a non−blank  CF_USER  value
   (the user−specified opacity correction filename).
     If the CNO−interpolation opacity files have not been read in, then this
   subroutine approximates the opacity effects of C,N,O,Ne interconversion by
   applying small negative and/or positive values of exC and exO ("excess−C,O")
   which may or may not be better than nothing.

   NOTE that interconversion of C,N,O,Ne via CNO−cycle burning changes slightly
   the value of Z that this subroutine will compute, for a given set of mass
   fractions of elements heavier than Ne: the total number density in C,N,O,Ne
   is constant, but the total mass in these elements changes.  Thus this
   subroutine OPAL_X_CNO_FU should only be used if one has read in a RANGE of
   Z−values (see opacity−reading subroutines below): if one has read in only a
   single Z−value, then OPAL_X_CNO_FU is likely to yield out−of−range Z values.

   Note that the subroutine SET_CNO_FILES( cf_hz, cf_c, cf_o, cf_n, cf_user )
   (discussed below) can be used to set the names of the opacity files that are
   used to get the opacity shifts for CNO−interconversion (files cf_hz through
   cf_n) and any user−specified opacity shifts (cf_user, relative to cf_hz).
   NOTE that the first four of these files (cf_hz, cf_c, cf_o, cf_n) are those
   used for CNO−interpolation, and should all have the SAME number fractions
   for the elements heavier than Ne in their compositions (only C,N,O,Ne should
   be interconverted in these CNO−interpolation files).


*** OPAL_F_CNOU( z, xh, exC, exO, slt, slr, fcn, fcon, fcnone, fu )  For users
    −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−  who want
```

to use their own methods to compute the metallicity Z, the "excess" carbon and oxygen mass fractions EXC and EXO, and CNO-interpolation factors FCN, FCON, and FCNONE (as well as any user-factor FU) from their composition.

This interface is similar to OPAL(z, xh, exC, exO, slt, slr) except for the added CNO/user-interpolation factors:

```
     fcn     Multiplies opacity differences between files cf_c and cf_hz
     fcon    Multiplies opacity differences between files cf_o and cf_hz
     fcnone  Multiplies opacity differences between files cf_n and cf_hz
     fu      Multiplies opacity differences between files cf_user and cf_hz
```

Note that the inputs supplied to the opacity-reading subroutines (described below) control whether CNO-interpolation and/or user-interpolation opacity files are read in. If no CNO-interpolation files were read in, then the values of fcn, fcon, fcnone are ignored; if no user-interpolation file was read in, then the value of fu is ignored.

NOTE that all the previous opacity-calculating interfaces above ultimately call the subroutine OPAL_F_XCON_CNOU (described further below); the values of Z, XH, EXC, EXO, SLT, SLR, FCN, FCON, FCNONE, FU actually used can be checked by calling the subroutine ASK_LAST_XCNOU described below.

```
*** ASK_LAST_XCNOU( Z, X, XC, XO, SLT, SLR, FCN, FCON, FCNONE, FU )
    ---------------------------------------------------------------
```

This subroutine just returns the last-used OPAL_F_CNOU input values: it can be used to check these values, rather than including common /x_opal_z/ in the calling program.

```
*** ASK_LAST_OPAC( OPACT, DOPACT, DOPACR, DOPACTD, FEDGE, FTREDGE, FZEDGE )
    -----------------------------------------------------------------------
```

This subroutine just returns the last-computed opacity values, taken from common /e_opal_z/ (as an alternative to including this common block in the calling program), as described above (under the OPAC and OPAL routines).

```
*** ASK_OPAL_Z_MIX( IMIX, XIZ, N_X, FNINZ, N_N )
    --------------------------------------------
```

For the mix specified by IMIX , this subroutine returns the components of the metallicity Z, in the arrays XIZ(N_X) and FNINZ(N_N) ; note that up to 19 components are returned, the ratios by mass and by number respectively of {C,N,O,Ne,Na,Mg,Al,Si,P,S,Cl,Ar,K,Ca,Ti,Cr,Mn,Fe,Ni}. Mix specs are:

```
   IMIX = 9 (actually, any IMIX > 5 or < -10) : return the actual mixture used
   IMIX = 1 : return the 'GN93hz' main solar mix
   IMIX = 2 : return the 'Alrd96a2' [O/Fe] = 0.3 mix
   IMIX = 3 : return the 'C95hz' [O/Fe] = 0.4 mix
   IMIX = 4 : return the 'W95hz' [O/Fe] = 0.5 mix
   IMIX = 5 : return the user-specified [O/Fe] > 0 mix
   IMIX = 0 : return the meteoritic mix
   IMIX = -1 : return the alternate main solar mix (default 'GS98hz')
   IMIX = -2 : return the corresponding version of 'Alrd96a2' [O/Fe] = 0.3 mix
   IMIX = -3 : return the corresponding version of 'C95hz' [O/Fe] = 0.4 mix
   IMIX = -4 : return the corresponding version of 'W95hz' [O/Fe] = 0.5 mix
   IMIX = -5 : return the corresponding user-specified [O/Fe] > 0 mix
   IMIX = -6 : return the "default-CNO" mix (for CNO-interpolation)
   IMIX = -7 : return the CtoN mix (for CNO-interpolation)
   IMIX = -8 : return the COtoN mix (for CNO-interpolation)
   IMIX = -9 : return the CNOtoNe mix (for CNO-interpolation)
   IMIX = -10 : return the user-defined nonCNO-interpolation mix
```

```
*** ASK_OPAL_MIX_WT( ATwt, Nwt, ATwtHHe, NHHe, ATz, Nz )  This subroutine
    ----------------------------------------------------  returns the atomic
  weights and nuclear charges used for the components of the metallicity Z:

    ATwt(Nwt) = real array returning up to 19 atomic weight values, as used
                  for {C,N,O,Ne,Na,Mg,Al,Si,P,S,Cl,Ar,K,Ca,Ti,Cr,Mn,Fe,Ni};
                  if Nwt < 19 then only Nwt values will be returned, and
                  if Nwt < 1 then no atomic weight values will be returned.
    ATwtHHe(NHHe) = real array returning up to 2 atomic weight values, as used
                      for H and He.
    ATz(Nz) = real array returning up to 19 atomic numbers (nuclear charges),
               for {C,N,O,Ne,Na,Mg,Al,Si,P,S,Cl,Ar,K,Ca,Ti,Cr,Mn,Fe,Ni}.


             ================================================
             The subroutines that read in the OPAL opacities:
             ================================================


*** READ_OPAL_DUMP( IU, CF_D )  If a binary opacity file was created at some
    --------------------------  earlier time by DUMP_OPAL_OPAC below, then
  this subroutine can read it in again; note that ALL opacities and user
  settings are restored from this unformatted binary file.
  NOTE that the only advantage of this is speed: reading such an opacity
  dumpfile is MUCH faster than using READ_BASIC_OPAL_OPAC (or its alternates).

    IU = integer Fortran unit number: from 1 to 99, and not equal to 5 or 6;
             negative value means "use previous/default value" (if no positive
             Fortran unit number was ever given earlier, the default is 23).
    CF_D = character string: name of opacity dumpfile to be read in (INCLUDING
             THE DIRECTORY, if it is not a local file).  NOTE that this file
             must exist, and must not be in compressed form.

----[NOTE that the format of this dumpfile is INCONSISTENT with those produced
  by program versions between FEBRUARY 1, 2004 and SEPTEMBER 6, 2006; any such
  dumpfiles must be re-created by the present version.]


*** DUMP_OPAL_OPAC( IU, CF_D )  AFTER you have read in the opacities (from the
    --------------------------  plain-text OPAL opacity files) via one of the
  subroutines below, this subroutine can be used to dump an unformatted binary
  file of the current opacities and user settings (just as read in) for future
  re-use by the subroutine READ_OPAL_DUMP above.  Note that if you have read
  in conductive and/or molecular opacities as well (e.g., via the subroutine
  READ_EXTENDED_OPAC below), then these will also be stored in the dumpfile.

    IU = integer Fortran unit number: from 1 to 99, and not equal to 5 or 6;
             negative value means "use previous/default value".
    CF_D = character string: name of opacity dumpfile to be created (INCLUDING
             THE DIRECTORY, if it is not a local file); this file will be
             overwritten if it already exists.

  The size of this opacity dumpfile will depend on the number of Z-values
  stored; typically, 'z14xcotrin21.f' will store the maximum number of 14, for
  a size of about 24 Mb (while 'z5xcotrin21.f' can store at most 5, for 9 Mb;
  'z1xcotrin21.f' will always store 1, for 2.7 Mb) [if no molecular opacities
  were read in, these sizes will be 23 Mb, 8 Mb, and 1.7 Mb, respectively].

  [But if you read opacities using  READZEXCO  (rather than one of the simpler
  subroutines  READ_BASIC_OPAL_OPAC  or  READ_EXTENDED_OPAC  described below),
  then the input argument  NZIN  determines how many Z-values are used.]
```

```
*** SET_OPAL_DIR( cdirin )  The input character variable  cdirin  can be used
    ----------------------  to specify the directory where the OPAL-opacity
  ASCII files will subsequently be looked for (default is the local directory,
  which can also be specified by supplying a blank argument to SET_OPAL_DIR).
  Note: the total length of the directory name MUST NOT exceed 246 characters.

  Example:      call set_opal_dir( ’/home/username/opal_directory/’ )

  OR: to specify "look in local directory":      call set_opal_dir( ’ ’ )

  OR: for a local subdirectory:      call set_opal_dir( ’opal_directory/’ )


*** SET_MOL_DIR( cdir_mol )  The input character variable  cdir_mol  specifies
    -----------------------  the directory where the molecular-opacity ASCII
  files will subsequently be looked for (blank input means "use the default":
  i.e., try both the OPAL directory and the local directory).


*** SET_COND_DIR( cdir_cond )  The input character variable  cdir_cond  gives
    -------------------------  the directory where conductive-opacity ASCII
  files will subsequently be looked for (blank input means "use the default":
  i.e., try both the OPAL directory and the local directory).


*** SET_OPAL_LIST_LEVEL( LIST_LEVEL )  This subroutine can be used to make the
    ---------------------------------  opacity-reading routines print out the
  ASCII opacity files that are actually read in (this can be useful when one
  uses a subroutine such as READ_EXTENDED_OPAC to read in the "best available"
  Ferguson molecular opacities, for example):

    LIST_LEVEL = integer flag to determine whether names of the ASCII opacity
                   files that are read in are printed out as they are read in
                   (note that binary opacity dump-files are NEVER printed).
                 The input value of  LIST_LEVEL  has the following effect:
                    0 : (default) NO listing (printout) of opacity file names.
                    1 : print the first Type 2 OPAL opacity (Gz???.x??) file,
                          the first of each Ferguson molecular opacity case
                          (or Alexander file), any conductive opacity file(s),
                          and any Type 1 OPAL opacity file(s) (e.g., ’GN93hz’,
                          ’GS98hz’, ’AGS04hz’, ’W95hz’, ’AGS04hz_Ofe.5_W95’,
                          ’GS98hz.CtoN’, ’GS98hz.COtoN’, ’GS98hz.CNOtoNe’...).
                  > 1 : up to  LIST_LEVEL  Type 2 OPAL opacity files will be
                            printed, and up to  LIST_LEVEL  of each Ferguson
                            case read in (this is seldom useful, but for
                            example one might use LIST_LEVEL of 9999 to
                            print out EVERY opacity file that is read in).


*** SET_OPAL_LIST_UNIT( LIST_IU )  Set the integer Fortran unit number for the
    -----------------------------  above list output (default is 6, i.e., the
  standard output).  An input value of 5, of more than 99, or less than 0,
  will be reset to the default of 6 (standard output).  [Note that if the
  value of  LIST_IU  coincides with one of the Fortran units used to actually
  read in an opacity file, a warning will be printed to the standard output,
  and all subsequent list output will be printed to the standard output.]


*** READ_BASIC_OPAL_OPAC( IU, Z, CF_HZ, OFEBRACK, CF_OFE )  Read in the basic
    ------------------------------------------------------  OPAL opacities,
  possibly for non-solar alpha-element abundances (i.e., non-zero [O/Fe]):
    IU = integer Fortran unit number: from 7 to 96; note IU through IU+3 may
           be used (negative value means "use previous/default value").  Note
           that values from 0 to 6 will be reset to 7, and values greater than
           96 will be reset to 96.
    Z = (real) "typical" metal-mass-fraction value; opacities will be read in
          for as wide a range around Z as possible (negative means "use the
          default value of 0.02").  For the version  z14xcotrin21.f  of these
          subroutines, the value of Z is irrelevant, as ALL of the available
          metallicities will be read in (from 0.0 to 0.1).  For  z5xcotrin21.f
          metallicities within about a factor of two of Z will be available;
          for  z1xcotrin21.f  on the the single metallicity Z will be read in.
    CF_HZ = character variable (or string constant) giving the "main" opacity
              file to read in, typically 'GN93hz', 'GS98hz', or 'AGS04hz' (but
              one can specify any Type-1 OPAL opacity file having a solar mix;
              blank means "use the default of 'GN93hz'").  NOTE that not only
              will this file  CF_HZ  be read in, but ALSO the file 'GN93hz'
              and the 40 Type-2 files (Gz???.x??).
    OFEBRACK = (real) value of [O/Fe] = log{ (n_O/n_Fe) / (n_O/n_Fe)_Sun } ;
                 if this is non-zero, then opacities are interpolated between
                 the file  CF_HZ  and the file  CF_OFE  to get opacities for
                 the given [O/Fe] value.
    CF_OFE = character variable (or string constant) giving the name of the
               OPAL opacity file with non-zero [O/Fe]; if OFEBRACK = 0.0, then
               this argument is ignored (blank means "use the default": for
               CF_HZ = 'GN93hz' the default is CF_OFE = 'W95hz', otherwise the
               default is to append '_Ofe.5_W95' to the value of CF_HZ).

 ----- NOTE that if the opacity files on disk are in compressed form (suffix
        '.gz', '.Z', or '.bz2'), then they will be uncompressed (by gunzip,
        uncompress, or bunzip2, respectively), read in, and compressed again;
        however, this slows down the input and MAY LEAD TO INPUT ERRORS if
        more than one instance of this program is running.


*** READ_EXTENDED_OPAC(IU,Z,CF_HZ,OFEBRACK,CF_OFE,I_MOL,I_COND,I_CNO,CF_USER)
    -------------------------------------------------------------------------
  This is an extended version of the above subroutine READ_BASIC_OPAL_OPAC:

    IU, Z, CF_HZ, OFEBRACK, CF_OFE : as in READ_BASIC_OPAL_OPAC above.
    I_MOL = flag controlling whether the low-temperature molecular opacities
              will be used.  The most useful values are as follows:
            I_MOL = 0 : do not read in any molecular opacities
            I_MOL = 1 : look for molecular opacities for the same composition
                          as the OPAL opacities (if no matching molecular set
                          of opacities can be found, look for any available
                          set of molecular opacities).  If  OFEBRACK  is not
                          zero, try to add opacity-shifts to get opacities
                          appropriate for this input [O/Fe] value.
            I_MOL = -1 : read the molecular opacities, but do not use them to
                           extend the OPAL opacities to low T when using the
                           OPAL opacity-calculating subroutines described
                           above [the molecular opacities can be accessed by
                           the subroutine KAP_MOL, described further below].
            I_MOL = 21, 31, 41, ... 121 : read ONLY the corresponding set of
                           Fergson 2005 molecular opacities (21=GN93, 31=GS98,
                           41=L03, 51=AGS04, 61=S92, 71=S92AE, 81=GS98-.2,
                           91=GS98+.2, 101=GS98+.4, 111=GS98+.6, 121=GS98+.8).
            I_MOL = 991 : read ONLY the molecular opacities specified by a
                      previous call to SET_FERG_USER (described further below).
          ------[A more-detailed dexcription of I_MOL than most users will
                 need follows the descriptions of the other input variables.]
    I_COND = flag controlling whether the high-density conductive opacities
               will be read in and/or used:
                 0 = do not use these high-RHO conductive opacities at all.
                 1 = read these opacities in, and use them to extend the OPAL
                       opacities to high RHO when using opacity-calculating
                       subroutines described above.  By default, the file
                       'condall06.d' (Potekhin et al. 2006) is used; if not
                       found, 'condall06' is tried; if not found, then the OLD
                       file 'Condopac' (Hubbard & Lampe 1969) is used; if none
                       of these are found, it is a fatal error.
                 2 = same as 1, except that only the Potekhin et al. (2006)
                       file ('condall06.d' or 'condall06') is looked for.
                -1 = same as 1, but do not use conductive opacities to extend
                       the OPAL opacities when using the opacity-calculating
                       subroutines described above (they can be accessed by
                       the subroutine KAPCOND, described further below).
                -2 = same as -1, except that only the Potekhin et al. (2006)
                       file ('condall06.d' or 'condall06') is looked for.
    I_CNO = flag controlling whether the varied-CNO opacities will be used:
              0 = these varied-CNO opacities will neither be read in nor used.
              1 = these varied-CNO opacities will be read in from files with
                    names given by appending '.CtoN', '.COtoN', and '.CNOtoNe'
                    to the name given by the input  CF_HZ  , and will be
                    available to use with the opacity-calculating subroutine
                    OPAL_X_CNO_FU (or OPAL_F_CNOU) described above.
    CF_USER = character variable (or string constant) giving the name of a
                user-specified Type-1 OPAL opacity file to read in, for use
                with OPAL_X_CNO_FU (or OPAL_F_CNOU); a blank input string
                means that no such file will be read in or used.

 ----- A more detailed description of  I_MOL  and its allowed values
                               (MOST USERS WILL NOT NEED TO BOTHER WITH THIS):
    I_MOL = flag controlling whether the low-temperature molecular opacities
              will be used.  IF IT IS NON-ZERO, then if possible Ferguson et
              al. (2005) molecular opacities MATCHING the OPAL opacity file
              will be read in; IF no such matching files are found, then all
              possible molecular opacity files are checked for, and the first
              one found is read in -- looking first for a user-specified case
              (see SET_FERG_USER below), next for a case specified by a prior
              call to the subroutine SET_ALEX_FILE (see below), next for the
              cases AGS04, L03, GS98, GN93, S92, S92AE, GS98-.2 ... GS98+.8;
              if none of these are found, the Alexander & Ferguson (1994)
              opacities will be read in from the file 'Alexopac'; if not even
              this file is found, then it is a fatal error.
           --- If one of the non-alpha-enhanced Ferguson et al. 2005 cases was
              read in, and  abs(I_MOL) < 10  and the input value of  OFEBRACK
              is non-zero, then an attempt will be made to read in files so as
              to obtain opacity-shifts to yield opacities for this specified
              [O/Fe] value, using the GS98 GS98-.2 ... GS98+.8 cases if these
              are available, or if not then the S92 and S92AE cases are tried.
           --- The input value of  I_MOL  also has the following effects:
            For the (DEFAULT) Ferguson et al. 2005 case:
                0 = do not use low-T molecular opacities at all.
                1 = the molecular opacities are read in, and will be used
                      to extend the OPAL opacities to low T when using the
                      opacity-calculating subroutines described above;
                      the edge factor FKAEDGE will be unity for XZCO < 0.1
                      (where  XZCO = Z + exC + exO  is the total metal mass
                      fraction), and FKAEDGE will go to
                      zero for XZCO > 0.12;
                2 = same as I_MOL = 1
                3 = same as I_MOL = 1
                4 = similar to I_MOL = 1, but set edge factor FKAEDGE to zero
                      for too much "excess-CO":  abs(exC) + abs(exO) > CO_lim
                      where CO_lim = 0.2 * max( Z , 0.0001 , Z + exC + exO )
                      [this will not be done if you just call the subroutine
                      KAPFERG (which does not know what Z, exC, and exO are),
                      but it WILL be done for any other subroutine, including
                      KAP_MOL, KAPFERG, and any calls to molecular opacity
                      routines from OPAL-opacity calculating subroutines].
            For the (OLD) Alexander & Ferguson 1994 case:
                0 = do not use low-T molecular opacities at all.
                1 = the molecular opacities are read in; arbitrary values of
                      XZCO  will be allowed at X = 0, and  XZCO < 0.1  will be
                      allowed at all X (the edge factor FKAEDGE will go to
                      zero for XZCO > 0.15 at X > 0.03).
                2 = similar to I_MOL = 1, but always reduce the edge factor
                      FKAEDGE to zero as XZCO increases from 0.1 to 0.15,
                      even for X = 0.
                3 = similar to I_MOL = 2, but restrict to regions where the
                      Alexander tables were available (0.0001 < XZCO < 0.03,
                      the original table range), and set the edge factor
                      FKAEDGE to zero for  XZCO < 0.00005  or XZCO > 0.05 .
                4 = similar to I_MOL = 3, but set edge factor FKAEDGE to zero
                      for too much "excess-CO".
            For both types of molecular opacities:
               -1 = read in the molecular opacities, but do not use them to
                      extend the OPAL opacities to low T when using the OPAL
                      opacity-calculating subroutines described above [the
                      molecular opacities can be accessed by the subroutine
                      KAP_MOL (or KAPFERG or KAPALEX), described further
                      below].  Otherwise, same as I_MOL = 1.
               -2 = same as I_MOL = -1, with Z-constraints as for I_MOL = 2
               -3 = same as I_MOL = -1, with Z-constraints as for I_MOL = 3
               -4 = same as I_MOL = -1, with Z-constraints as for I_MOL = 4
            Other input values of  I_MOL  allow one to specify precisely which
            Ferguson et al. 2005 case to read in (note that for the following
            cases, NO [O/Fe]-molecular-opacity-shifts will be done):
               11 thru 14, or -11 thru -14 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson et al. 2005 case specified
                      by a previous call to SET_ALEX_FILE is looked for.
               21 thru 24, or -21 thru -24 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GN93 case is looked for ...
               31 thru 34, or -31 thru -34 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GS98 case is looked for ...
               41 thru 44, or -41 thru -44 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson L03 case is looked for ...
               51 thru 54, or -51 thru -54 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson AGS04 case is looked for ...
               61 thru 64, or -61 thru -64 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson S92 case is looked for ...
               71 thru 74, or -71 thru -74 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson S92AE case is looked for ...
               81 thru 84, or -81 thru -84 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GS98-.2 case is looked for ...
               91 thru 94, or -91 thru -94 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GS98+.2 case is looked for ...
              101 thru 104 or -101 thru -104 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GS98+.4 case is looked for ...
              111 thru 114 or -111 thru -114 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GS98+.6 case is looked for ...
              121 thru 124 or -121 thru -124 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only the Ferguson GS98+.8 case is looked for ...
              991 thru 994 or -991 thru -994 : same as 1 thru 4 or -1 thru -4,
                      EXCEPT: only a user-specified Ferguson case is looked
                      for (as previously set by calling SET_FERG_USER).
```

```
*******************************************************************************



  ***************************************************************************
  SUBROUTINES TO CONTROL THE DETAILS: MOST USERS WILL NOT NEED TO USE THESE:
  ***************************************************************************



******* NOTE that the opacity-reading subroutines described above call various
  versions of the subroutines described below; for example, calling
    READ_BASIC_OPAL_OPAC( IU, Z, CF_HZ, OFEBRACK, CF_OFE )
  does the equivalent of:
      CALL SET_ALTMIX_MAIN_FILE( CF_HZ )
      CALL SET_OFE_FILE( CF_OFE )
      CALL SET_ALTMIX_OFE_FILE( CF_OFE )
      CALL READZEXCO( -9, -1.0, z, -1.0, khighz, IU, OFEBRACK )
  with a value of  khighz  determined by the inputs  CF_HZ  and  OFEBRACK;
  likewise, calling
    READ_EXTENDED_OPAC(IU,Z,CF_HZ,OFEBRACK,CF_OFE,I_MOL,I_COND,I_CNO,CF_USER)
  does the equivalent of:
      CALL SET_ALTMIX_MAIN_FILE( CF_HZ )
      CALL SET_OFE_FILE( CF_OFE )
      CALL SET_ALTMIX_OFE_FILE( CF_OFE )
      CALL SET_CNO_FILES( ' ', ' ', ' ', ' ', CF_USER )
      CALL READZEXCO( -9, -1.0, Z, -1.0, khighz, IU, OFEBRACK )
      CALL READ_BEST_MOL( IU, I_MOL, CF_HZ, OFEBRACK )
      IF ( IABS( I_COND ) .GT. 1 ) THEN
         CALL READ_COND_POT( IU )
      ELSE IF ( I_COND .NE. 0 ) THEN
         CALL READCOND( IU )
      ENDIF
      CALL SET_COND_USE( MIN(1,MAX(0,I_COND)), 99 )
  where the value of  khighz  is determined by the various inputs.

  Likewise, the opacity-calculating subroutines  OPAC, OPAL, OPAL_X_CNO_FU,
  and OPAL_F_CNOU  will call  OPAL_F_XCON_CNOU,  which then calls one or more
  of the subroutines  OPAL_K_ONLY, KAPCOND, KAPFERG, KAPALEX  described below.


    =========================================================================
    Other subroutines that interpolate among the OPAL or other opacities:
    =========================================================================


*** OPAL_K_ONLY( Z, XH, EXC, EXO, SLT, SLR, FCN, FCON, FCNONE, FU )  This is
    ----------------------------------------------------------------  almost
  the same as OPAL_F_CNOU, with the same meaning for the inputs, EXCEPT that
  it will only return the OPAL opacities (i.e., it ignores molecular opacities
  and conductive opacities, whether or not these have been read in).


*** KAPFERG( slt, slr, xh, z, exC, exO, FLKA, DLKATR, DLKARO, DLKAT, FKAEDGE )
    --------------------------------------------------------------------------
  This routine interpolates and returns the Ferguson et al. 2005 molecular
  opacity (provided that a set of these opacities has been read in)  [NOTE
  that if these Ferguson et al. 2005 molecular opacities are available, then
  the Alexander & Ferguson 1994 molecular opacities are not, and vice versa]:

    slt, slr, xh, z, exC, exO = (real) input values, as in subroutine OPAL.
    FLKA = (real) variable returning Log(kappa), equivalent to OPACT
    DLKATR = (real) variable returning dLog(kappa)/dLog(T) at constant R,
               equivalent to DOPACT
    DLKARO = (real) variable returning dLog(kappa)/dLog(RHO) at constant T,
               equivalent to DOPACR
    DLKAT = (real) variable returning dLog(kappa)/dLog(T) at constant density,
              equivalent to DOPACTD
    FKAEDGE = (real) variable returning the edge factor, equivalent to FEDGE


*** CACHEFERG( ksto, xh, xzco )  This subroutine caches Z- and X-interpolated
    ---------------------------  Ferguson et al. 2005 molecular opacity values
  for future use [if a LARGE number of low-termperature opacities must be
  evaluated with the same Z and X values, this can speed up the evaluation].
    ksto = (integer) storage index, 1 or 2, of the cache storage area (only
             two sets of Z- and X-interpolated opacities can be cached, at
             X1,Z1 and X2,Z2).
    xh = X, (real) hydrogen mass fraction to use.
    xzco = Z + exC + exO, (real) total metal mass fraction to use.


*** KAP_MOL( slt, slr, xh, z, exC, exO, FLKA, DLKATR, DLKARO, DLKAT, FKAEDGE )
    --------------------------------------------------------------------------
  This routine just calls either KAPFERG (see above) or KAPALEX (see below),
  depending on which type of molecular opacities are available.


*** KAPALEX( FLT, FLRO, X, XZCO, FLKA, DLKAT, DLKARO, FKAEDGE )  This routine
    -----------------------------------------------------------  interpolates
  and returns the Alexander & Ferguson 1994 molecular opacities (provided that
  these opacities have been read in from the file ’Alexopac’)  [NOTE that if
  these Alexander & Ferguson 1994 molecular opacities are available, then the
  Ferguson et al. 2005 molecular opacities are not, and vice versa]:

    FLT = log(T) = SLT + 6.0
    FLRO = log(RHO) = SLR + 3.0 * SLT
    X = hydrogen mass fraction (the same as XH above)
    XZCO = Z + exC + exO = 1 - X - Y = the total mass fraction of elements
             heavier than helium (note that the Alexander & Ferguson 1994
             opacities are available only as a function of temperature,
             density, hydrogen abundance, and metallicity)
    FLKA = (real) variable returning Log(kappa), equivalent to OPACT
    DLKAT = (real) variable returning dLog(kappa)/dLog(T) at constant density,
              equivalent to DOPACTD
    DLKARO = (real) variable returning dLog(kappa)/dLog(RHO) at constant T,
               equivalent to DOPACR
    FKAEDGE = (real) variable returning the edge factor, equivalent to FEDGE


*** ASK_LAST_ALEX_EDGE( FKAEDGE, FTRA, FTRA_LO, FTRA_HI, FZKAEDGE )  To give
    ---------------------------------------------------------------  access to
  the detailed edge factors for the molecular opacities (which yielded the
  final overall value of FKAEDGE, as returned by KAPFERG or KAPALEX above):

    FKAEDGE = real variable returning the final overall edge factor; it is
                defined as  FKAEDGE = FTRA * FZKAEDGE  .
    FTRA = real variable returning the temperature-density edge factor, which
             is defined as  FTRA = min( FTRA_LO , FTRA_HI )  .
    FTRA_LO = real variable returning the edge factor for the low T, low RHO
                edges of the molecular opacity tables.
    FTRA_HI = real variable returning the edge factor for the high T, high RHO
                edges of the molecular opacity tables.
    FZKAEDGE = real variable returning the molecular Z-edge factor, in the
                 manner determined by the value of  KALEX  as set by the
                 subroutine  READ_EXTENDED_OPAC  (or SET_ALEX_USE below).
```

```
*** SET_ALEX_USE( KALEX )  This allows the user to control whether molecular
    ---------------------  opacities that were read in will actually be used,
  and how; it may be called at any time to change this behavior:

    KALEX = integer input flag controlling handling of molecular opacities:
              if KALEX > 0, then the OPAL-opacity-calculating subroutines will
                               use the molecular opacities (if they have been
                               read in) as an extension to lower T;
              if KALEX < 0 or KALEX = 0, then the OPAL opacity-calculating
                               subroutines will ignore the molecular opacities
                               whether or not they have been read in (but the
                               subroutine KAPFERG or KAPALEX will still work).
              Specific values for KALEX control the molecular-opacity Z-edge
              factor, whether called from OPAL-opacity-calculating subroutines
              or from the subroutines KAPFERG or KAPALEX:
               1 : (DEFAULT) For Ferguson et al. 2005 opacities, the Z-edge
                    factor FZKAEDGE is unity for XZCO < 0.10 at all X, and
                    goes to zero for XZCO > 0.12; but for the Alexander 1994
                    opacities, FZKAEDGE is unity for X = 0 at all XZCO and for
                    XZCO < 0.10 at all X, and it goes to zero at XZCO > 0.15
                    and X > 0.03 (note that XZCO = 1 - X - Y = Z + exC + exO).
               2 : For Ferguson et al. 2005, same meaning as KALEX = 1; but
                    for Alexander 1994, FZKAEDGE goes to zero for XZCO > 0.15
                    at all X.
               3 : For Ferguson et al. 2005, same meaning as KALEX = 1; but
                    for Alexander 1994, restrict to the regions where tables
                    were available (not extrapolated or otherwise created):
                      FZKAEDGE = 1.0 for 0.0001 < XZCO < 0.03, and goes to
                      FZKAEDGE = 0.0 for XZCO < 0.00005 or XZCO > 0.05 .
               4 : FOR THE OPAL-OPACITY-CALCULATING SUBROUTINES and KAPFERG:
                    for BOTH the Ferguson et al. 2005 and the Alexander 1994
                    cases: multiply the FZKAEDGE value from case 3 above by a
                    factor that is unity if  exC  and  exO  are both zero,
                    and which goes to zero if  abs(exC) + abs(exO) > CO_lim
                    for  CO_lim = 0.2 * max( Z , 0.0001 , Z + exC + exO ) .
                    [Note that this constraint cannot be applied in KAPALEX,
                    which does not know the separate values of Z, exC, exO
                    (but only their sum XZCO); thus, for calls to KAPALEX
                    itself, KALEX = 4 has the same effect as KALEX = 3.]
               0 : Same as KALEX = -1.
              -1 thru -4 : molecular opacities should NOT be used by the OPAL
                    opacity-calculating routines, even if they are read in;
                    same as 1 thru 4 when KAPFERG is called explicitly;
                    same as 1 thru 3 when KAPALEX is called explicitly.
              -9 : Set KALEX to its default value of 1.
             -99 : Leave the current value of KALEX unchanged.


*** ASK_ALEX_USE( KALEX, KALEX_AVAIL, ITYPE )  This subroutine returns the
    -----------------------------------------  current value of these integer
  flags that control the use of molecular opacities (see SET_ALEX_USE above):

    KALEX = integer variable returning the value of KALEX as in SET_ALEX_USE.
    KALEX_AVAIL = integer variable returning a flag value telling whether (and
                    which) molecular opacities are actually available:
                      0 : not available (molecular opacities never read in).
                      1 : Alexander & Ferguson 1994 molecular opacities are
                            available (the file 'Alexopac' has been read in):
                            subroutine KAPALEX can be used to return Alexander
                            opacities, and if KALEX > 0 then these will be
                            used to extend the OPAL opacities to low T.
                      2 : some set of Ferguson et al. 2005 molecular opacities
                            is available; subroutine KAPFERG can be used to
                            return these, and if KALEX > 0 then these will be
                            used to extend the OPAL opacities to low T.
    ITYPE = integer variable returning a flag value, whose meaning depends on
               which type of molecular opacities are available:
             ITYPE = ITYPE_FERG for Ferguson et al. 2005 molecular opacities:
              the index of the (last) type of Ferguson opacities that were
              read in (note that if several types of Ferguson opacities were
              combined using READ_ADD_FERG or ADDFILE_FERG, the index refers
              to the last type to be read in and combined):
                   1 = case specified by CFILE_ALEX (see SET_ALEX_FILE below),
                   2 = GN93 case,      3 = GS98 case,      4 = L03 case,
                   5 = AGS04 case,     6 = S92 case,       7 = S92AE case,
                   8 = GS98-.2 case,   9 = GS98+.2 case,  10 = GS98+.4 case,
                  11 = GS98+.4 case,  12 = GS98+.8 case,
                  99 = user-specified case (see SET_FERG_USER above).
             ITYPE = IRHOSW for Alexander & Ferguson 1994 molecular opacities:
              integer variable returning a flag value telling whether there is
              a density switchover between OPAL and Alexander opacity tables
              (as well as the temperature switchover which must be present)
              [see subroutine SET_ALEX_DO_RHOSW described further below; note
              Ferguson et al. 2005 opacities NEVER have a density switchover].


*** KAPCOND( FLRO, FLT, X, Y, XCN, XON, XNEHEAVY, FMUAINV, FMUEINV,
             ZSQBAR, IDER, FLKC, FLKCT, FLKCRO, FKCEDGE, FKCOK )
    ---------------------------------------------------------------
  This subroutine calculates and returns the conductive opacities.

  If the file ’condall06.d’ was read in, use the Potekhin et al. 2006 tables;
  whether RMS or linear mean ionic charge is used depends on IDER (see below),
  where  Zion(rms)    = (<z^2)^0.5 = sqrt( ZSQBAR / FMUAINV )  ,
         Zion(linear) = <z>        = FMUEINV / FMUAINV  .

  Alternatively, if the file ’Condopac’ was read in instead, use the Hubbard &
  Lampe 1969 tables, possibly extended and modified by formulae of Itoh et al.
  1983 and Mitake et al. 1984, depending on the value of IDER (see below).

    FLRO = log(RHO) = SLR + 3.0 * SLT
    FLT = log(T) = SLT + 6.0
    X = hydrogen mass fraction (the same as XH above).
    Y = helium mass fraction.
    XCN = Xc + Xn / 2 = effective total carbon mass fraction.
    XON = Xo + Xn / 2 = effective total oxygen mass fraction.
    XNEHEAVY = Xne + Xheavy = mass fraction of elements heavier than oxygen
                 [by definition,  X + Y + XCN + XON + XNEHEAVY = 1].
    FMUAINV = 1 / mu_A = SUM_i{ X_i / A_i }  (for all elements i in the mix --
                note that X_i is the mass fraction and A_i the atomic weight);
                if FMUAINV < 0.0, then an approximate value is calculated from
                the input values of X, Y, XCN, XON, and XNEHEAVY.
    FMUEINV = 1 / mu_e = SUM_i{ X_i * z_i / A_i }  (for all elements i in the
                mix -- note that z_i is the atomic number of element i);
                if FMUEINV < 0.0, then an approximate value is calculated.
    ZSQBAR = <z^2> / mu_A = SUM_i{ X_i * (z_i)^2 / A_i }  (for all elements i
                in the mix); this is the mean square ionic charge <z^2>
                divided by mu_A (i.e., multiplied by FMUAINV) for the mixture;
                if ZSQBAR < 0.0, then an approximate value is calculated.
    IDER = flag controlling which conductive opacities are returned.
             For the (DEFAULT) Potekhin 2006 (’condall06.d’) case:
               1 : bi-quadratic interp, Zion = (<z^2)^0.5, with derivatives;
                     NOTE:  Zion(rms) = (<z^2)^0.5 = sqrt( ZSQBAR / FMUAINV );
                     if it is necessary to extrapolate in log(Zion), do linear
                     extrapolation using the (quadratic) slope at the edge of
                     the matrix; if it is necessary to extrapolate in logT or
                     logRHO, do linear extrapolation using the slope of the 2
                     points at the edge of the matrix (this linear formula is
                     mixed with a quadratic if one moves inside the edge of
                     the matrix; full bi-quadratic occurs one grid-spacing
                     inside the edge).
               0 : bi-quadratic interp, Zion = (<z^2)^0.5, no derivatives.
              -1 : use the Potekhin-website formulae, Zion = (<z^2)^0.5, with
                     derivatives [uses linear interpolation in log(Zion)];
                     extrapolate linearly using slope at edge of matrix.
                     (This cubic interpolation from the Potekhin-website file
                     'condint.f' is slower than bi-quadratic interpolation,
                     and may be slightly noisier when using single precision
                     real variables, but yields similar results).
              -2 : bi-quadratic interp, Zion = <z>, with derivatives;
                     NOTE:  Zion(linear) = <z> = FMUEINV / FMUAINV .
              -3 : bi-quadratic interp, Zion = <z>, no derivatives.
              -4 : Potekhin-website formulae, Zion = <z>, with derivatives.
             For the (OLD) H&L ('Condopac') case:
               1 : combine as necessary, with derivatives: use "H&L" at lower
                     density, "Itoh" at higher density, with switchover region
               0 : combine as necessary (no derivatives).
              -1 : get "Itoh" conductive opacities only (no derivatives)
              -2 : get possibly-modified "H&L" opacities only (no derivatives)
              -3 : get "Itoh" conductive opacities only (with derivatives)
              -4 : get possibly-modified "H&L" only (with derivatives)
          NOTE: "H&L" is W. B. Hubbard & M. Lampe 1969, ApJS 163, 297; "Itoh"
          is N. Itoh, S. Mitake, H. Iyetomi, & S. Ichimaru 1983, ApJ 273, 774
          plus S. Mitake, S. Ichimaru, & N. Itoh 1984, ApJ 277, 375.
    FLKC = (real) variable returning Log(Kappa_cond), equivalent to OPACT
    FLKCT = (real) variable returning dLog(kappa_cond)/dLog(T) at constant
              density, equivalent to DOPACTD (note that a value of 0.0 will be
              returned if IDER indicates derivatives should not be calculated)
    FLKCRO = (real) variable returning dLog(kappa_cond)/dLog(RHO) at constant
               temperature, equivalent to DOPACR
    FKCEDGE = (real) variable returning the edge factor, equivalent to FEDGE
                or FTREDGE (note that conductive opacities have no metallicity
                restrictions); but a negative value is returned for regions
                more than one grid-spacing off the edge of the matrix, and a
                value of -99999.0 is returned if conductive opacities are not
                available.
    FKCOK = (real) variable returning a somewhat less restrictive edge factor,
              which is a better indicator of whether the returned conductive
              opacity is good enough to combine with (or extend) the radiative
              OPAL opacity.


*** KAP_COND_POT( FLRO, FLT, ZION, IDER, FLKC, FLKCT, FLKCRO, FKCEDGE, FKCOK )
    --------------------------------------------------------------------------
  This subroutine calculates and returns the Potekhin et al. (2006) conductive
  opacities (if they were not read in, it returns, indicating out-of-range).

    FLRO = log(RHO) = SLR + 3.0 * SLT      (density, logarithm to the base 10)
    FLT = log(T) = SLT + 6.0           (temperature, logarithm to the base 10)
    ZION = Zion, the mean ionic charge (the user must calculate this for their
             mixture, e.g. the rms nuclear charge weighted by number density).
             It is a fatal error if Zion < 0.1 or Zion > 130.
    IDER = flag controlling which conductive opacities are returned:
              1 : bi-quadratic interpolation, with derivatives; switched over
                    to linear extrapolation at matrix edge.
              0 : bi-quadratic interpolation, no derivatives.
             -1 : use the Potekhin-website formulae, with derivatives [linear
                    interpolation in log(Zion), cubic interpolation in logT,
                    logRHO]; linear extrapolation via slope at matrix edge.
    FLKC, FLKCT, FLKCRO, FKCEDGE, FKCOK : (real) variables returning values as
                                            in KAPCOND above.


*** SET_COND_USE( KCOND, KREPLACE_ITOH )  This subroutine allows the user to
    ------------------------------------  control the way conductive opacities
  will be handled; it may be called at any time to change this behavior:

    KCOND = integer input flag controlling handling of conductive opacities:
              1 : (DEFAULT): if a conductive-opacity file has been read in,
                    then the conductive opacity will be combined with the
                    OPAL opacity to yield the overall opacity value when OPAL
                    opacity routines are called (and also used to extend the
                    OPAL opacity to higher density).
              0 : even if a conductive-opacity file has been read in, the
                    conductive opacities will be ignored when computing OPAL
                    opacities (i.e., the conductive opacities can only be
                    accessed by the subroutine KAPCOND described above).
             -9 : reset to the default value of 1.
            -99 : leave the present stored value of this flag unchanged.
    KREPLACE_ITOH = integer input flag controlling how conductive opacities
                      are calculated:
                       99 : (DEFAULT) use the most recent/best conductive
                              opacities available.
                       -9 : reset to the default value of 99.
                      -99 : leave the present stored value unchanged.
                      For Potekhin 2006:
                        2 : use bi-quadratic interpolation in logRHO, logT,
                              and log(Zion), where Zion = (<z^2>)^0.5 (rms).
                        1 : use the formulae from the website: i.e., cubic
                              interpolation in logRHO, linear interpolation in
                              log(Zion), and cubic interpolation in logT,
                              where Zion = (<z^2>)^0.5 (rms).
                        0 : use bi-quadratic interpolation in logRHO, logT,
                              and log(Zion), where Zion = <z> (linear mean).
                       -1 : use the formulae from the website: i.e., cubic
                              interpolation in logRHO, linear interpolation in
                              log(Zion), and cubic interpolation in logT,
                              where Zion = <z> (linear mean).
                      For H&L 1969:
                        2 : reserved (at present, means the same as 0).
                        1 : reserved (at present, means the same as 0).
                        0 : use the 1983/1984 ’Itoh’ conductive opacities to
                              modify and extend the ’H&L’ ones.
                       -1 : use only these ’H&L’ conductive opacities, not the
                              1983/1984 ’Itoh’ ones.


*** ASK_COND_USE( KCOND, KCOND_AVAIL, KREPLACE_ITOH )  This returns the values
    -------------------------------------------------  of these integer flags
  that control the use of conductive opacities (see SET_COND_USE above):

    KCOND = integer variable returning the value of KCOND as in SET_COND_USE.
    KCOND_AVAIL = integer variable returning returning a flag value telling
                    whether the conductive opacities are actually available:
                     0 : not available (NO conductive opacities have ever
                           been read in).
                     1 : H&L 1969 conductive opacities have been read in.
                     2 : Potekhin 2006 conductive opacities have been read in.
    KREPLACE_ITOH = integer variable returning the value of KREPLACE_ITOH as
                      in SET_COND_USE above.

  [NOTE: if KCOND_AVAIL > 0, then the subroutine KAPCOND can be used to get
  conductive opacities, and if KCOND > 0 then these conductive opacities will
  be used to extend the OPAL opacities to high RHO.]



         =============================================================
         Other subroutines that may be used when reading in opacities:
         =============================================================


*** SET_ALEX_FILE( CFILEALEX, I_FULL_PATH )  This subroutine allows the user
    ---------------------------------------  to specify a molecular opacity
  file whose name has been changed, or which is in a different directory.
----NOTE: ANY DIRECTORY that you specified via SET_MOL_DIR will be DISCARDED.

    CFILEALEX = character variable or string constant giving a non-default
                    name for the molecular opacity file(s); NOTE that if the
                    input string  CFILEALEX  ends with a slash ('/'),
                    then it will be treated as a directory (or subdirectory)
                    specification and the default filename will be appended.
                    For Ferguson et al. (2005) molecular opacities, the
                    (optional) directory specification may be followed by a
                    filename beginning-part (as in SET_FERG_USER just below).
    I_FULL_PATH = integer flag indicating how this should be interpreted:
                    0 : (DEFAULT): look for molecular opacity file(s) in the
                          same directory as the OPAL opacity files (or a
                          subdirectory thereof, if CFILEALEX contains a
                          subdirectory specification).
                    1 : just look for file(s) called (or whose names start
                          with) CFILEALEX (i.e., either the file(s) are in
                          the local directory, or else the string CFILEALEX
                          includes any required directory pathname).


*** SET_FERG_USER( CBEG_FERG )  This subroutine allows the user to set the
    --------------------------  beginning-part of the names for a non-default
  set of the Ferguson et al. (2005) molecular opacities:

    CBEG_FERG = character variable or string constant giving the beginning of
                  the file names, including at the end any dot ('.') that
                  separates beginning part from the X-value in the filename
                  (for the  default mixes,  CBEG_FERG  would be  'g'  'g98.'
                  'l03.'  'ags04.'  's92.'  's92ae.'  'gs98-.2.'  'gs98+.2.'
                  'gs98+.4.'  'gs98+.6.'  or  'gs98+.8.'  for GN93, GS98, L03,
                  AGS04, S92, S92AE, GS98-.2, GS98+.2, GS98+.4, GS98+.6, or
                  GS98+.8 mixes, respectively --- note that the GN93 mix is
                  the only one of these that would not have the separating dot
                  '.' at the end of the CBEG_FERG value).
                [Note that this mix-specification may optionally be preceded
                  by a subdirectory-specification, e.g., 'f05_gs98/g98.']


*** ASK_FERG_USER( CBEG_FERG )  This subroutine returns the value stored by
    --------------------------  calling SET_FERG_USER as described just above.

    CBEG_FERG = character variable, to return the file-name beginning.


*** SET_FERG_ACC( IACC )  This subroutine allows one to change the accuracy
    --------------------  with which the Ferguson et al. (2005) molecular
  opacities are interpolated (YOU SHOULD NOT DO SO):

    IACC = (integer) flag value controlling the accuracy:
```

```
             1 : (DEFAULT): when there is a large opacity jump between two
                   adjacent grid-points at low temperature (logT < 3.48),
                   avoid the large overshoot that would result just outside
                   these grid-points from the usual biquadratic interpolation.
             0 : (NOT RECOMMENDED!): just use usual biquadratic interpolation,
                   which yields somewhat faster computations at logT < 3.48,
                   but which can lead to errors of order a factor of 2, or
                   even occasionally of more than an order of magnitude.


*** ASK_FERG_ACC( IACC )  This subroutine returns the value of IACC presently
    --------------------  in use, as per the subroutine SET_FERG_ACC above.

    IACC = integer variable to return the value of the flag IACC.


*** READ_BEST_MOL( IU, I_MOL, CF_HZ, OFEBRACK )  If I_MOL is non-zero, this
    -------------------------------------------  subroutine tries to read the
  molecular opacities that best match the OPAL opacities specified by CF_HZ
  and OFEBRACK (see subroutine READ_EXTENDED_OPAC further above):

    IU, I_MOL, CF_HZ, OFEBRACK : same as in READ_EXTENDED_OPAC further above.


*** READFERG( IU )  This subroutine reads in the default set of Ferguson et
    --------------  al. (2005) molecular opacities.  Ths subroutine just calls
  the subroutine READ_ADD_FERG( IU, 0.0, 1.0, 0 ) below:

    IU = (integer) Fortran unit number to use; a value of 5 or of 6 will be
           reset to 7, and a negative value means "use the default of 23 or
           the previous specified Fortran unit".


*** ADDFILE_FERG( IU, F_STO, F_READ, CFILEFERG, I_FULL_PATH )  This subroutine
    ---------------------------------------------------------  reads in a set
  of Ferguson et al. (2005) molecular opacities, as specified by the user,
  optionally combining them with a previously-input set (it is a fatal error
  if the specified Ferguson opacity files are not found):

    IU = (integer) Fortran unit number (as in READFERG above).
    F_STO = (real) value by which to multiply already-stored logKappa values,
              before adding the logKappa values newly read in; use F_STO = 0.0
              to just read in a new set of opacities (if no Ferguson molecular
              opacities have been read in previously, then the input value of
              F_STO is ignored, and a value of 0.0 is used).
    F_READ = (real) value by which to multiply the logKappa values read in
                 from the specified files; typically, F_READ = 1.0 is combined
                 with F_STO = 0.0 to just read in a new set of opacities (a
                 value of F_READ = 0.0 returns without reading anything).
    CFILEFERG = character variable or string constant giving the beginning of
                    the Ferguson et al. (2005) molecular opacity filenames (as
                    in SET_ALEX_FILE or SET_FERG_USER above); this specifies
                    which set to use.
    I_FULL_PATH = integer flag indicating how this should be interpreted (as
                    in SET_ALEX_FILE above):
                    0 : (DEFAULT): look for molecular opacity file(s) in the
                          same directory as the OPAL opacity files (or a
                          subdirectory thereof, if CFILEFERG contains a
                          subdirectory specification).
                    1 : just look for files whose names start with CFILEFERG
                          (i.e., either the files are in the local directory,
                          or else the string CFILEFERG includes any required
                          directory pathname).
```

```
*** READ_ADD_FERG( IU, F_STO, F_READ, KTYPE_FERG )  This subroutine reads in
    ----------------------------------------------  a set of either default or
  non−default Ferguson et al. (2005) molecular opacities, as specified by the
  user, optionally combining them with a previously−input set (it is a fatal
  error if no Ferguson opacity files are found):

    IU, F_STO, F_READ = as in subroutine ADDFILE_FERG just above.
    KTYPE_FERG = flag indicating which molecular opacities should be read in:
                   0 : look for the first available set of Ferguson opacities,
                         in the order 99,1,5,4,3,2,6,7,8,... of cases below:
                   1 : read the Ferguson mix that was specified previously by
                         a call to SET_ALEX_FILE or ADDFILE_FERG (see above).
                   2 : read the GN93 Ferguson mix ’g’ −−− if this is not found
                         in the directory specified by SET_OPAL_DIR above (or
                         by SET_ALEX_FILE above), look in the subdirectory
                         ’f05_g93/’ as well.
                   3 : read the GS98 Ferguson mix ’g98.’ (also look in the
                         subdirectory ’f05_gs98/’)
                   4 : read the L03 Ferguson mix ’l03.’ (also look in the
                         subdirectory ’f05_l03/’)
                   5 : read the AGS04 Ferguson mix ’ags04.’ (also look in the
                         subdirectory ’f05_ags04/’)
                   6 : read the S92 Ferguson mix ’s92.’ (also look in the
                         subdirectory ’f05_s92/’)
                   7 : read the S92AE Ferguson mix ’s92ae.’ (also look in the
                         subdirectory ’f05_s92ae/’)
                   8 : read the GS98−.2 Ferguson mix ’gs98−.2.’ (also look in
                         the subdirectory ’f05_gs98−.2/’)
                   9 : read the GS98+.2 Ferguson mix ’gs98+.2.’ (also look in
                         the subdirectory ’f05_gs98+.2/’)
                  10 : read the GS98+.4 Ferguson mix ’gs98+.4.’ (also look in
                         the subdirectory ’f05_gs98+.4/’)
                  11 : read the GS98+.6 Ferguson mix ’gs98+.6.’ (also look in
                         the subdirectory ’f05_gs98+.6/’)
                  12 : read the GS98+.8 Ferguson mix ’gs98+.8.’ (also look in
                         the subdirectory ’f05_gs98+.8/’)
                  99 : (actually, any value larger than 12): read the Ferguson
                         mix specified previously by a call to SET_FERG_USER.
                  −1 to −99 : first try the case  abs(KTYPE_FERG)  ; if it is
                                not found, behave as if case 0 was specified.


*** READALEX( IU )  This subroutine just reads in the file ’Alexopac’ (with
    --------------  Alexander & Ferguson 1994 molecular opacities), using the
  Fortran unit IU (provided that this file has not already been read in).


*** SET_COND_FILE( CFILECOND, I_FULL_PATH )  This subroutine allows the user
    ---------------------------------------  to specify a which conductive
  opacity file should be read in, and/or a file whose name has been changed,
  or which is in a different directory.
----NOTE: ANY DIRECTORY that you specified via SET_COND_DIR will be DISCARDED.
----NOTE: Calling this subroutine SET_COND_FILE causes any previously−stored
             conductive opacities to be DISCARDED at once.

    CFILECOND = character variable or string constant giving the revised name
    I_FULL_PATH = integer flag indicating how this should be interpreted:
                    0 : (DEFAULT): look in same directory as for OPAL files.
                    1 : just look for a file called CFILECOND.
```

```
*** SET_COND_INFLAGS( KCOND_FIX, KCOND_GAP, KCOND_HAVE )  This subroutine sets
    ----------------------------------------------------  flags controlling
  how the 'H&L' conductive opacity tables are modified when they are read in
  (this will have an effect only if called BEFORE you read in the opacities):

    KCOND_FIX = integer input flag controlling an error-fix:
                  1 : (DEFAULT): remove an anomalously low Hydrogen conductive
                        opacity value (at logT = 5.4, logRHO = -0.5).
                  0 : do not do this (NOT RECOMMENDED).
                 -9 : reset to the default value of 1.
                -99 : leave the present stored value unchanged.
    KCOND_GAP = integer input flag controlling the "gap" in the 'H&L' tables,
                  i.e., the density region where they do not extend to such
                  low temperatures as they do at higher or lower densities:
                    1 : (DEFAULT): high-density 'Itoh' conductive opacities
                          are used in this "gap" to improve the 'H&L' opacity
                          there (beyond the edge of the 'H&L' tables) in order
                          to improve the H&L-to-Itoh switchover at increasing
                          density and low temperature (note that no input H&L
                          values are modified, only the extrapolation region).
                    0 : this is not done: H&L tables are just extrapolated.
                   -9 : reset to the default value of 1.
                  -99 : leave the present stored value unchanged.
    KCOND_HAVE = integer input flag controlling whether the file 'Condopac'
                   (or 'condall06.d') will be read in again:
                     1 : (DEFAULT): it will NOT be read in again (unless it
                           has never been read in at all), even if you call
                           the subroutine READCOND a second time, unless you
                           change the filename by calling SET_COND_FILE.
                     0 : re-read the file 'Condopac' the next time you call
                           the subroutine READCOND.
                   -9 : if and only if you have changed the stored value(s) of
                          KCOND_FIX or KCOND_GAP, re-read the file 'Condopac'
                          the next time you call the subroutine READCOND.
                  -99 : (equivalent to specifying the default value of 1).


*** ASK_COND_INFLAGS( KC_FIX, KC_GAP, KC_FIX_N, KC_GAP_N )  This subroutine
    ------------------------------------------------------  returns integer
  flag values (as may be set by the subroutine SET_COND_INFLAGS above):

    KC_FIX = integer variable returning the value of KCOND_FIX that was used
               the last time 'Condopac' was read in.
    KC_GAP = integer variable returning the value of KCOND_GAP that was used
               the last time 'Condopac' was read in.
    KC_FIX_N = integer variable returning the value of KCOND_FIX that will be
                 used the NEXT time 'Condopac' is read in.
    KC_GAP_N = integer variable returning the value of KCOND_GAP that will be
                 used the NEXT time 'Condopac' is read in.


*** READ_COND_POT( IU )  This subroutine just reads in the Potekhin 2006
    -------------------  conductive opacity file 'condall06.d', using Fortran
  unit IU (provided that it has not already been read in).


*** READCOND( IU )  This subroutine just reads in either the Potekhin 2006
    --------------  conductive opacity file 'condall06.d' or the older H&L
  file 'Condopac', using Fortran unit IU (provided that conductive opacities
  have not already been read in).


*** SET_OFE_FILE( cfileofe )  The input character variable  cfileofe  can be
```

```
    −−−−−−−−−−−−−−−−−−−−−−−  used to set a user−specified filename containing
  non−CO−rich opacities with [O/Fe] > 0.0, similar to ’Alrd96a2’, ’C95hz’, or
  ’W95hz’.  Only the filename (not the directory pathname) should be specified
  and the length of the filename MUST NOT exceed 8 characters.  This filename
  is only used in the case  khighz = 5, 15, 25, or 35  (see READZEXCO below).


*** SET_ALTMIX_OFE_FILE( cfileofe )  The input character variable  cfileofe
    −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−  can be used to set a user−specified
  filename containing non−CO−rich GS98 opacities with [O/Fe] > 0.0; the length
  of the name is only restricted by the fact that filename plus OPAL directory
  name cannot exceed 255 characters in total.  This filename is only used in
  the case  khighz = −5, −15, −25, or −35  (see READZEXCO below).


*** SET_ALTMIX_MAIN_FILE( cfile_hz )  The input character variable  cfile_hz
    −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−  can be used to replace the alternate
  main file ’GS98hz’ with a user−specified filename; this new file will be
  assumed to have [O/Fe] = 0.0, i.e., a solar mix.  The length of the name is
  only restricted by the fact that filename plus OPAL directory name cannot
  exceed 255 characters in total.  This filename is used whenever khighz < 0
  (see description in READZEXCO below); note that khighz = −2, −3, and −4
  should NOT be used subsequently, unless this file replacing ’GS98hz’ also
  has the Grevesse & Sauval 1998 mix.


*** SET_CNO_FILES( cf_m, cf_c, cf_o, cf_n, cf_user )  This subroutine can be
    −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−  used to set the five
  alternate opacity filenames that can be used to obtain the CNO−interpolation
  opacity shifts (plus a user−specified case); the input character variables
  are only restricted by the fact that filename plus OPAL directory name can’t
  exceed 255 characters in total (blank values mean "use default filenames"):

    cf_m = a standard opacity file (with standard composition); the default
             used in READZEXCO is ’GN93hz’ if  khighz > 0,  or else  cfile_hz
             (e.g., ’GS98hz’: see SET_ALTMIX_MAIN_FILE above) if  khighz < 0
    cf_c = an opacity file where most or all C (by number) has been converted
             to N; the default filename is  cf_hz  with ’.CtoN’ appended
    cf_o = an opacity file where most/all of both C and O have been converted
             to N; the default filename is  cf_hz  with ’.COtoN’ appended
    cf_n = an opacity file where all CNO has been converted to Ne; the default
             filename is  cf_hz  with ’.CNOtoNe’ appended
    cf_user = a user−specified opacity file, with a composition that can be
                arbitrarily different from that in the file cf_hz; the default
                filename is  cf_hz  with ’.user’ appended

  NOTE that the first four of these files (cf_m, cf_c, cf_o, cf_n) should all
  have the SAME number fractions for the elements heavier than Ne (only C,N,O,
  Ne should be interconverted in these CNO−interpolation files).

  Note that as long as the files cf_m, cf_c, cf_o, and cf_n have compositions
  that are not linearly dependent (or close to it) in the 3−dimensional space
  of interconversion of C, N, O, and Ne, the CNO−interpolation should still
  work correctly.  However, it has been tested only for the specific case
  described above.


*** SET_CNO_EXT( ie, ce_hz, ce_c, ce_o, ce_n, ce_u )  This subroutine can
    −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−  be used to change the
  default extensions for the CNO−interpolation files (which are used to decide
  on the default CNO−filenames); this can be used as an alternative to calling
  the subroutine SET_CNO_FILES.  As noted above, these default extensions are
  ’ ’ , ’.CtoN’ , ’.COtoN’ , ’.CNOtoNe’ , ’.user’  respectively.  Calling this
  subroutine sets these default extensions to the values of the character
  variables  ce_hz, ce_c, ce_o, ce_n, ce_u  respectively.  The integer  ie
  controls the meaning of a blank input character value:
    ie = 0 : in all cases, a blank input character value means do not change
             the corresponding default extension (UNLESS YOU HAVE SOME VERY
             PECULIAR NAMING CONVENTIONS FOR CNO FILES, THIS IS THE ONLY
             CASE YOU ARE LIKELY TO NEED)
    ie = 1 : if  ce_hz  is blank, reset the corresponding extension to blank
             (i.e., the default filename remains  'GN93hz'  or  cfile_hz);
             but if any of the other input character values are blank, do
             not change these other corresponding default extension
    ie = 2 : if  ce_c  is blank, reset the corresponding extension to blank
    ie = 3 : if  ce_o  is blank, reset the corresponding extension to blank
    ie = 4 : if  ce_n  is blank, reset the corresponding extension to blank
    ie = 5 : if  ce_u  is blank, reset the corresponding extension to blank
    ie = 6 thru 10 : same as ie = 1 thru 5, respectively
  Note that an extension longer than 80 characters cannot be accommodated.


*** SET_METEOR_MIX_FILE( cfilemet )  The input character variable  cfilemet
    -------------------------------  can be used to set a user-specified
  filename containing a meteoritic mixture.  This file is used ONLY to set
  the components of Z for a meteoritic mixture, which are available to the
  user but are not used in calculating any opacities.


*** ASK_OPAL_FILE_USED( ITYPE, CF_USED )  This subroutine returns the name of
    ------------------------------------  the specified type of opacity file
 that was (or will be) used to read in the opacities:

    ITYPE = integer input flag controlling which file name is returned:
              -1 : the name of the OPAL directory CDIRIN.
               0 : meteoritic-mix file CFILEMET (this file may not exist).
               1 : the main mix file CF_HZ.
               2 : molecular opacities file CFILEALEX, including directory.
               3 : conductive opacities file CFILECOND, including directory.
               4 : the file 'GN93hz' (needed to correct Gz???x?? files).
               5 : the non-zero-[O/Fe] file CF_OFE.
               6 : the CNO-interpolation main mix file CF_M.
               7 : the CNO-interpolation C --> N file CF_C.
               8 : the CNO-interpolation CO --> N file CF_O.
               9 : the CNO-interpolation CNO --> Ne file CF_N.
              10 : the user-mix file CF_USER.
    CF_USED = character variable returning the filename specified by ITYPE.


*** ASK_KHIGHZ_OFE( KHIGHZ_USED, OFEBRACK_USED )  This subroutine returns the
    --------------------------------------------  values that were used when
  the subroutine READZEXCO was called to read in the opacities (directly or
  via READ_BASIC_OPAL_OPAC or READ_EXTENDED OPAC):

    KHIGHZ_USED = integer variable returning the value of KHIGHZ (see below).
    OFEBRACK_USED = real variable returning the value of [O/Fe] that was used.


*** READZEXCO( Nzin, Zlo, Z, Zhi, khighz, iulow, ofebrack )  This subroutine
    -------------------------------------------------------  is used to read
  in the OPAL opacity files, allowing the user to control whether and how
  opacities will subsequently be interpolated in Z.  Note that a new set of
  opacities (at a new Z-range or Z-value) can be read in at any time.

  The controlling input variables are:
    Nzin      INTEGER: the number of metallicity values Z_i to be stored,
                for subsequent Z−interpolation when OPAL or OPAC is called;
                this is discussed just below.  Nzin = −9 means "use the widest
                available Z−range around the input Z−value".
    Zlo       SINGLE−PRECISION REAL: the lowest metallicity value that will
                be required; this is discussed just below.
    Z         SINGLE−PRECISION REAL: the "typical" or central metallicity
                value; this is discussed just below.
    Zhi       SINGLE−PRECISION REAL: the highest metallicity value that will
                be required; this is discussed just below.
    khighz    INTEGER: controls the use of the C=O=0.0 opacity file ’GN93hz’
                (and/or its equivalents with newer mixes, such as ’GS98hz’ and
                ’AGS04hz’), and of the similar files having [O/Fe] > 0.0:
                 khighz = 0: use of the file ’GN93hz’ is prevented; only for
                              X < 0.75 is accurate X−interpolation available.
                 khighz = 1: the file ’GN93hz’ is used to obtain opacities for
                              the C=O=0.0 mixes (i.e., opacities with better
                              Z−interpolation), including the added X−values
                              X={0.2,0.5,0.8,0.9,0.95,1−Z} (i.e., allowing
                              accurate X−interpolation up to X = 1−Z); for
                              the mixes with C+O > 0.0, corresponding opacity
                              shifts are applied, for consistency.
                 khighz = 2: file ’Alrd96a2’ with [O/Fe] = 0.3 \  is used in
                 khighz = 3: file ’C95hz’ with [O/Fe] = 0.4     } addition to
                 khighz = 4: file ’W95hz’ with [O/Fe] = 0.5    /  ’GN93hz’,
                              if READZEXCO was called with a non−zero value of
                              ofebrack, in order to interpolate in the excess
                              oxygen/alpha−element enrichment [O/Fe].
                 khighz = 5: the name of a file with non−zero [O/Fe] must have
                              been set already, by calling the subroutine
                              SET_OFE_FILE described below; it will be used
                              instead of ’Alrd96a2’, ’C95hz’, or ’W95hz’ when
                              interpolating in [O/Fe] (its [O/Fe] value will
                              be computed when it is read in; if it actually
                              has [O/Fe] = 0.0, the resulting behavior is not
                              defined and will surely be erroneous).
                 khighz = −1 thru −5: similar to khighz = 1 thru 5, except
                              that a different set of OPAL opacity files is
                              used, defining a different set of heavy−element
                              abundances to comprise the solar metallicity Z.
                              THE OLD FILE ’GN93hz’ IS STILL REQUIRED AS WELL,
                              but the opacities now stored are those from the
                              new file with the same format (called ’GS98hz’,
                              by default), and this is the composition that
                              is assigned a value of [O/Fe] = 0.0; khighz = −2
                              thru −5 likewise implies the use of files with
                              [O/Fe] > 0.0 relative to the mix in ’GS98hz’: by
                              default ’GS98hz_OFe.3_Alrd96a2’ at [O/Fe] = 0.3,
                              ’GS98hz_OFe.4_C95’ at 0.4, ’GS98hz_OFe.5_W95’
                              at 0.5, or user−defined for khighz = −5 via the
                              subroutine SET_ALTMIX_OFE_FILE (see below).  The
                              main alternate solar−composition [O/Fe]=0.0 file
                              name can be changed from ’GS98hz’ by calling the
                              subroutine SET_ALTMIX_MAIN_FILE (see below); if
                              this is done, khighz = −2 thru −4 should not be
                              used subsequently unless the replacement main
                              file still uses the Grevesse & Sauval 1998 mix;
                              rather, if one wishes opacities with a non−zero
                              [O/Fe] value, one should do something such as:
                               call set_altmix_main_file( ’AGS04hz’ )
                               call set_altmix_ofe_file( ’AGS04hz_OFe.5_W95’ )
                              and then use khighz = −5.
                 khighz = −11 thru −15,  OR
```

```
                        11 thru 15: same as khighz = −1 thru −5 OR 1 thru 5,
                            except that CNO−interpolation opacity files are
                            read in (if possible: uses the filenames "CF_HZ,
                            CF_C, CF_O, CF_N" that can be set by calling
                            SET_CNO_FILES: see below)
               khighz = −21 thru −25,  OR
                        21 thru 25: same as khighz = −1 thru −5 OR 1 thru 5,
                            except that a user−specified (OPAL) opacity
                            interpolation file is read in (if possible: uses
                            the filenames "CF_HZ, CF_U" that can be set by
                            calling SET_CNO_FILES: see below)
               khighz = −31 thru −35,  OR
                        31 thru 35: same as khighz = −1 thru −5 OR 1 thru 5,
                            except that BOTH the CNO− and user−specified
                            opacity interpolation files are read in (if
                            possible)
  iulow     INTEGER: the beginning Fortran unit number for reading opacity
              files; Fortran units  iulow  through  iulow + 3  may be used.
              Zero or negative iulow values mean "use previous (or default)
              value", where the default value is iulow = 23.  A fatal error
              will result if iulow < 7 or iulow > 96 (unless you have set
              the error level to 0, in which case these values are ignored).
              (Note: unless the user explicitly calls READZEXCO, READEXCO,
              READCO, or READ_OPAL_DUMP, the default−setup call to READZEXCO
              in OPAL will be executed, yielding the default iulow of 23).
  ofebrack  SINGLE−PRECISION REAL: the value of [O/Fe], the logarithmic
              oxygen (or alpha−element) enhancement factor, relative to the
              Sun:  ofebrack = log10{ (O_mix/Fe_mix) / (O_sun/Fe_sun) } ,
              where O_mix, Fe_mix, O_sun, and Fe_sun are number densities.
              If  khighz = 0, 1, or −1, then ofebrack is ignored; otherwise,
              READZEXCO interpolates (or extrapolates) log(Kappa) linearly
              between mix 1 (or −1) and mix  mod(khighz,10) , interpolation
              factors being such as to yield the desired [O/Fe] by combining
              these mixes.  Note: ’GN93hz’ has [O/Fe] = 0.0 by definition,
              ’Alrd96a2’ has [O/Fe] = 0.3, ’C95hz’ has [O/Fe] = 0.4, and
              ’W95hz’ has [O/Fe] = 0.5, but they have different patterns of
              enhancements for elements other than oxygen; their elemental
              abundances and the corresponding opacity shifts are discussed
              further below.

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
Discussion of  Nzin, Zlo, Z, Zhi  in calling the above subroutine READZEXCO:
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

Z−interpolation of opacity is actually carried out in terms of log(Z+0.001).
The maximum number of z−values that can be stored (to interpolate among) is
given by the value of the constant  NZ  in the parameter statements that
begin as  "parameter ( nz=" .  The maximum sensible value is NZ = 14, which
requires about 22.7 Mb of opacity matrix storage space.  Other reasonable
values include NZ = 8 (13.0 Mb) and NZ = 5 (8.1 Mb); a value of NZ = 3 (4.9
Mb) still allows quadratic Z−interpolation, while NZ = 2 (3.2 Mb) allows
only linear interpolation in log(Z+0.001); for NZ = 1 (1.6 Mb), the program
behaves much the same as the earlier version of MAY 28, 1999 (or as if the
subroutines READCO or READEXCO were used instead of READZEXCO).

If you have reduced the error−checking level to 0 (using SET_ERR_CHECK),
then the input value of  Nzin  will be decreased if necessary so that it
does not exceed  NZ , the maximum available number of Z−storage values;
otherwise, a value of  Nzin > NZ  or of  Nzin < 1  is a fatal error, EXCEPT
THAT  Nzin = −9  means "use maximum possible value, i.e., as if Nzin = NZ".
If necessary, the subroutine ASK_Z_LIMITS can be called to check the value
of this hard−wired limit  NZ  , or the subroutine ASK_Z_USE can be called
after READZEXCO to check the number of Z−values actually used  (see below).
```

Values of Zlo, Z, Zhi that are within 1.E−6 of one of the file z−values { 0.0, 0.0001, 0.0003, 0.001, 0.002, 0.004, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.1 } are generally reset to be exactly equal to that value (except that only the range [−1.E−6 , 1.E−8] is reset to be exactly zero). Any value of Zlo, Z, or Zhi greater than 0.1 is always a fatal error.

Significantly negative Z−values (below −1.E−6) mean "use default values":
- if all three of Zlo, Z, Zhi are negative, then Z is reset to 0.02;
- if only Z is negative, then it is reset to lie between Zlo and Zhi ;
- if Zlo and/or Zhi is negative, then the negative value(s) will be reset to "reasonable" values, according to the values of Nzin and Z.

If Nzin = 2 , then the stored z−values are given by the input values of Zlo and Zhi ; if both of these are negative, then a range +/− 10% in Z is used; if only one of them is negative, a total range of 20% in Z is used, or more if the remaining interval [Zlo,Z] or [Z,Zhi] is larger than this. The minimum allowed range is 1% in Z, or delta−Z = 1.E−5 for small Z values; this is a fatal error, unless you have reduced the error−checking level to 0, in which case the program quietly uses this lower limit. Likewise too large a range: Zlo < min{ 0.6 * Zhi , Zhi − 0.0002 } is a fatal error, unless you have reduced the error−checking level to 0, in which case the ONLY UPPER LIMIT on the linear Z−interpolation range is that it must lie within [0.0,0.1]; BEWARE that large ranges yield inaccurate interpolation.

If Nzin > 2 , then from the set of eight "largest−allowed−spacing" Z−values { z1=0.0, z2=0.001, z3=0.004, z4=0.01, z5=0.02, z6=0.03, z7=0.05, z8=0.1 }, choose the largest z_J and the smallest z_K such that z_J is no greater than Zlo and z_K is no less than Zhi; it is then a fatal error if Nzin < K − J , i.e., if the Z−range is too large for the given value of Nzin (unless of course you have reduced the error−checking level to 0, in which case arbitarily large ranges are accepted: BEWARE!). Also...

If Nzin > 2 , then: if a set of Nzin adjacent file z−values from the set { 0.0, 0.0001, 0.0003, 0.001, 0.002, 0.004, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.1 } encompasses the range [Zlo,Z,Zhi] , then such a set of Nzin z−values is used (as far as possible, it will be centered on Z): for example, for Nzin = 3 , input Z−values [Zlo,Z,Zhi] = [0.01,0.02,0.03] or [0.017,0.018,0.019] or [0.022,0.022,0.024] or [0.019,0.028,0.029] all yield { 0.02, 0.03, 0.04 }, while [0.021,0.028,0.029] yields { 0.02, 0.03, 0.04 }.

If Nzin = 3 and no set of 3 of the above file z−values will work, then the actual input values are used (except that, if the logarithmic interval from Zlo to Z is sufficiently different from that from Z to Zhi, the value of Z is reset to the logarithmic midpoint of Zlo and Zhi): for example, the input Z−value set [Zlo,Z,Zhi] = [0.012,0.024,0.04] yields { 0.012, 0.024, 0.04 }, while [0.012,0.015,0.04] yields { 0.012, 0.02208679, 0.04 }.

If Nzin > 3 and no set of Nzin of the above file z−values will work, then try whether a similar set that works can be obtained by removing (some of) the z−values that are not present in the C,O−rich OPAL opacity files Gz???.x?? , which are available at { 0.0, 0.001, 0.004, 0.01, 0.02, 0.03, 0.05, 0.1 }; if such a set is found (with somewhat larger z−intervals), then it is used. Otherwise, endpoints Zlo and Zhi are used, with remaining z−values equally spaced in log(Z+0.001) between these endpoints.

******* NOTE that if you have set the error−level to 0, then there is NO UPPER LIMIT on the maximum allowed Z−range (except that it must lie in the range [0.0,0.1] where OPAL opacities are available), and thus QUITE INACCURATE Z−interpolation will occur if the input Z−range [Zlo , Zhi] is relatively large and Nzin (or NZ) is relatively small.

One or two of the Type−1 OPAL files can be read in (as specified by khighz in your call to the opacity-reading subroutine READZEXCO). For example,

```
      call readzexco( 14, 0.0, -1.0, 0.1, 1, 23, 0.0 )
```

will read in opacities at all Z-values from 0.0 to 0.1, reading additional opacities from 'GN93hz' (due to the value of khighz=1); Fortran units 23 through 26 will be used for input, and the opacities will have [O/Fe]=0.0 (solar composition for the abundances comprising Z). For non-CO-rich cases, this allows slightly improved Z-interpolation (for Z < 0.12) and slightly improved X-interpolation (for 0.03 < X < 0.75); for high hydrogen abundances (X > 0.75), such as may result from diffusion (e.g., helium settling), the accuracy is GREATLY IMPROVED.

NOTE that only the version z14xcotrin21.f allows Nzin = 14 as in the above call; if one is using the less-memory-hogging version z5xcotrin21.f , then the above call would lead to a fatal error, unless you had reduced the error checking level, in which case it would lead to opacities being read in only for the Z-values Z = { 0.0, 0.00217, 0.00905, 0.0309, 0.1 }, yielding MUCH LESS ACCURATE Z-interpolation; USER BEWARE!!

For z5xcotrin21.f , only a small Z-range should be read in: for example,

```
      call readzexco( 5, -1.0, 0.02, -1.0, 4, 23, 0.45 )
```

reads opacities at Z = .004, .01, .02, .03, .04, using 'GN93hz' and 'W95hz' to obtain opacities at [O/Fe]=0.45 (the results are then as accurate as with the version z14xcotrin21.f for the restricted Z-range 0.01 < Z < 0.03: this is described in more detail below).

For z1xcotrin21.f , only a SINGLE Z-value can be read in; the opacities from the files will be interpolated in Z while being read in, if necessary.

To read opacities in the widest available Z-range around some metallicity Z that is compatible with accuracy, one would use Nzin = -9, e.g.,

```
      call readzexco( -9, -1.0, Z, -1.0, 11, 23, 0.0 )
```

This would read opacities with [O/Fe] = 0.0 in a range around Z that is determined by the value of NZ in the parameter statements in this file.

NOTE that if opacities have not already been read in, then the first time that an opacity-calculating subroutine is called (e.g., OPAC or OPAL), it will use "call readzexco(-9, -1.0, z, -1.0, 1, 23, 0.0)" to read in the opacities, i.e., basic opacities for the maximum reasonable Z-range (with [O/Fe] = 0.0 and no CNO-interpolation, only interpolation in "excess-C,O").

```
*** READCO( Z, kallrd, khighz, iulow )
    ----------------------------------
*** READEXCO( Z, kallrd, khighz, iulow, ofebrack )
    ----------------------------------------------
```

These subroutines behave the same as the previous version of MAY 28, 1999: opacities are read in at the SINGLE metallicity Z (interpolated among available OPAL opacity files, if necessary), but subsequently opacities are available only at this single metallicity, unless and until the user explicitly reads in a new set of opacities via another call to READCO, READEXCO, or READZEXCO. (For READCO, any positive value of khighz is set to 1, and any negative value to -1).

NOTE THAT the input INTEGER kallrd is ignored (it is included only for backward compatibility).

===

Subroutines used to control the switchover from OPAL to molecular opacities:

===

NOTE that the Ferguson and the Alexander switchover spcifications are entirely separate: only one will take effect at any time, namely, the one for which the corresponding molecular opacities have been read in.

```
*** SET_LOGT_SW_FERG( FLTSW_LO, FLTSW_HI )  Set the temperature-boundaries of
    --------------------------------------  the switchover region from OPAL to
  Ferguson et al. (2005) molecular opacities (this subroutine can be called
  at any time, to change the switchover temperatures):

    FLTSW_LO = lowest logT of switchover region, where only Ferguson is used:
                 -99.0 : leave current stored logT value unchanged.
                 -50.0 : use the default value (of logT = 4.2).
                 > -20 : use the input value FLTSW_LO, except that:
                           it will not be lower than 3.75 (lowest T for OPAL),
                           it will not exceed 4.45 (near highest Ferguson T).
    FLTSW_HI = highest logT of switchover region, where only OPAL is used:
                 -99.0 : leave current stored logT value unchanged.
                 -50.0 : use the default value (of logT = 4.4).
                 > -20 : use the input value FLTSW_HI, except that:
                           it will not exceed 4.50 (highest Ferguson T),
                           it will not be lower than 3.80 (near lowest OPAL).

  Note that in no case will  FLTSW_HI - FLTSW_LO < 0.05  be allowed to occur
  (the values will be moved further apart if this happens).


*** ASK_LOGT_SW_FERG( FLTSW_LO, FLTSW_HI )  Return the current Ferguson logT
    --------------------------------------  switchover values (as above).


*** SET_LOGT_SW_ALEX( FLTSW_LO, FLTSW_HI )  Set the temperature-boundaries of
    --------------------------------------  the switchover region from OPAL to
  Alexander & Ferguson 1994 molecular opacities (this subroutine can be called
  at any time, to change the switchover temperatures):

    FLTSW_LO = lowest logT of switchover region, where only Alexander is used:
                 -99.0 : leave current stored logT value unchanged.
                 -50.0 : use the default value (of logT = 3.87).
                 > -20 : use the input value FLTSW_LO, except that:
                           it will not be lower than 3.75 (lowest T for OPAL),
                           it will not exceed 4.05 (near highest Alexander T).
    FLTSW_HI = highest logT of switchover region, where only OPAL is used:
                 -99.0 : leave current stored logT value unchanged.
                 -50.0 : use the default value (of logT = 3.97).
                 > -20 : use the input value FLTSW_HI, except that:
                           it will not exceed 4.10 (highest Alexander T),
                           it will not be lower than 3.80 (near lowest OPAL).

  Note that in no case will  FLTSW_HI - FLTSW_LO < 0.05  be allowed to occur
  (the values will be moved further apart if this happens).


*** ASK_LOGT_SW_ALEX( FLTSW_LO, FLTSW_HI )  Return the current Alexander logT
    --------------------------------------  switchover values (as above).


*** SET_ALEX_DO_RHOSW( IRHOSW )  Set the flag controlling whether there is a
    ---------------------------  density-switchover too (in the temperature
  region where both OPAL and Alexander 1994 opacities are available, but where
  the OPAL opacities are available up to higher densities):

  IRHOSW = input integer flag:
              1 : (DEFAULT): use a density-switchover region too.
              0 : do not do so.
             -9 : use the default value of 1.
            -99 : leave the stored value unchanged.
```

```
*** SET_LOGRHO_SW_ALEX( FLRHOSW_LO, FLRHOSW_HI )  Set the density-boundaries
    --------------------------------------------  of the switchover region
  (for Alexander 1994 opacities) that is enabled/disabled by SET_ALEX_DO_RHOSW
  above (this can be called at any time, to change the switchover densities):

  FLRHOSW_LO = lowest logRHO of switchover region, where only Alexander used:
                 -99.0 : leave current stored logRHO value unchanged.
                 -50.0 : use the default value (of logRHO = -6.25).
                 > -20 : use the input value FLTSW_LO, except that it will lie
                           in the range  -14.0 < FLRHOSW_LO < -6.0  .
  FLRHOSW_HI = highest logRHO of switchover region, where only OPAL is used:
                 -99.0 : leave current stored logRHO value unchanged.
                 -50.0 : use the default value (of logRHO = -5.75).
                 > -20 : use the input value FLTSW_HI, except that it will lie
                           in the range  -13.75 < FLRHOSW_HI < -5.75  .

  Note that in no case will  FLRHOSW_HI - FLRHOSW_LO < 0.25  be allowed to
  occur (the values will be moved further apart if this happens).


*** ASK_LOGRHO_SW_ALEX( FLRHOSW_LO, FLRHOSW_HI )  Return current switchover
    --------------------------------------------  logRHO values (as above).


*** SET_LOGT_RHOSW_ALEX( FLTSW_R_LO, FLTSW_R_HI )  Set the temperature-bounds
    ---------------------------------------------  below which the Alexander-
  to-OPAL density-switchover described above does not take place (due to the
  low-temperature limit of the OPAL opacities):

    FLTSW_R_LO = lowest logT of switchover region, where only Alexander used:
                   -99.0 : leave current stored logT value unchanged.
                   -50.0 : use the default value (of logT = 3.70).
                   > -20 : use the input value FLTSW_LO, except that:
                             it will not be lower than 3.70 (the lowest
                               allowed OPAL-extrapolation logT value),
                             it will not exceed FLTSW_HI - 0.05 (see the
                               subroutine SET_LOGT_SW_ALEX above).
    FLTSW_R_HI = highest logT of switchover region, where only OPAL is used:
                   -99.0 : leave current stored logT value unchanged.
                   -50.0 : use the default value (of logT = 3.80).
                   > -20 : use the input value FLTSW_HI, except that:
                             it will not exceed FLTSW_HI,
                             it will not be lower than 3.75

  Note that in no case will  FLTSW_R_HI - FLTSW_R_LO < 0.05  be allowed to
  occur (the values will be moved further apart if this happens).


*** ASK_LOGT_RHOSW_ALEX( FLTSW_R_LO, FLTSW_R_HI )  Return density-switchover
    ---------------------------------------------  logT-bounds as above.


         ==========================================================
         The subroutines that control the X-interpolation accuracy:
         ==========================================================


*** SET_XHI( kxhi )  Set a flag telling whether or not to use the additional
    ---------------  ’GN93hz’ X-values for more accurate X-interpolation
  (provided they are available, i.e., ’GN93hz’ or ’GS98hz’ has been read in).
    If  kxhi = 2  , then a flag is set such that the ’GN93hz’ X-values will be
```

```
                      used whenever they are available (this is the DEFAULT
                      case, if you never call SET_XHI).  Note that only at
                      X > 0.03 will the 'GN93hz' X-values affect the resulting
                      interpolated opacity values.
    If  kxhi = 1  , then a flag is set such that the 'GN93hz' X-values will be
                      used whenever they are available, but ONLY for values of
                      X > 0.7 (this yields faster but slightly less accurate
                      X-interpolation for X < 0.7, while retaining accurate
                      opacities for large X-values up to X = 1-Z (such as may
                      arise from diffusive processes).
    If  kxhi = 0  , then a flag is set such that the 'GN93hz' X-values will
                      NOT be used, even when they are available (this results
                      in only slightly poorer X-interpolation for X < 0.75,
                      but yields wildly incorrect opacities for very large X
                      values, i.e., for X approaching 1-Z).
  Note that the 'GN93hz' X-values are available for X-interpolation ONLY if
  the 'GN93hz' file has been read in, i.e., if READZEXCO above has been called
  with a non-zero value of  khighz  among its input parameters.  Note also
  that, strictly, the 'GN93hz' X-values are defined only for non-CO-rich mixes
  (C=O=0.0); but corresponding opacity shifts are applied for consistency up
  to a CO-enhancement of C+O = 0.2, these shifts being reduced to zero as C+O
  increases from 0.2 to 0.3 and being ignored for C+O of 0.3 or more.


*** ASK_XHI( kxhi, kavail )  Returns INTEGER VARIABLE flags telling whether
    -----------------------  'GN93hz' X-values will be used, and whether they
  are actually available at the moment:
  Returns  kxhi  value as set most recently by SET_XHI above (i.e., returns
                         kxhi = 0, 1, or 2, with the same meaning as above);
                         if SET_XHI was never called, then returns  kxhi = 2 .
  Returns  kavail = 1  if the 'GN93hz' files have been read in, i.e., the
                         'GN93hz' X-values are available, and will be used for
                         X-interpolation if the value of  kxhi  so indicates.
           kavail = 0  if the 'GN93hz' file has NOT (yet) been read in,
                         i.e., the 'GN93hz' X-values are not available, and
                         can NOT be used for X-interpolation no matter what
                         value  kxhi  has (unless 'GN93hz' is read in later).


      ===============================================================
      The subroutines that control the CNO and/or user-interpolation:
      ===============================================================


*** SET_CNO_INTERP( kcno, kuser )  Set flags telling whether or not to use the
    -----------------------------  CNO/user-interpolation opacity shifts; by
  default, both are used, providing they are available (i.e., providing the
  relevant opacity files were read in: see flag  khighz  in READZEXCO above).
    If  kcno > 0 ,  then the CNO-interpolation opacity shifts will be used (if
                      available); otherwise, they will be ignored
    If  kuser > 0 ,  then the user-specified opacity shifts will be used (if
                       available); otherwise, they will be ignored


*** ASK_CNO_INTERP( kcno, kuser, kcno_avail, kuser_avail )  Returns INTEGER
    ------------------------------------------------------  VARIABLE flags to
  indicate whether CNO/user-interpolation opacity shifts will be used when
  obtaining opacities.
    Returns  kcno, kuser  as set by SET_CNO_INTERP above (or their default
                            values of 1 if SET_CNO_INTERP was never called)
    Returns  kcno_avail, kuser_avail  values of 1 if the corresponding opacity
                                        files have been read in, or 0 if not
```

```
         ==========================================================
         The subroutines that control the level of error-checking:
         ==========================================================


*** SET_ERR_CHECK( LEVEL )  This subroutine sets the error-checking level to
    ----------------------  the given (integer) input value LEVEL:
  Level = 0 : Only minimal error checking is performed on inputs.  A Z-value
                above 0.1 in the arguments to the opacity-reading subroutine
                READZEXCO is a fatal error, as is an inconsistent composition
                input to the opacity-calculating subroutines OPAC or OPAL;
                most other problematic input is handled or accepted silently,
                in a manner that ought to be reasonable (but no guarantees!).
  Level = 1 : This is the DEFAULT case (which will occur if you never call the
                subroutine SET_ERR_CHECK).  At this level, error-checking is
                performed on the arguments of the subroutine READZEXCO (which
                one calls to read in the opacities).  As described above in
                the discussion of  Nzin, Zlo, Z, Zhi  , it is a fatal error
                if  Nzin < 1 or if Nzin is too large (exceeding the available
                number of Z-storage spaces).  It is also a fatal error if the
                Z-range [Zlo,Zhi] is too small or too large.  At this level,
                a warning will be issued if you call SET_ALTMIX_MAIN_FILE and
                subsequently use khighz = -2, -3, or -4 (see above), or if any
                filename exceeds 255 characters (or an extension exceeds 80).
  Level = 2 : At this level of error-checking, in addition:  If the arguments
                to OPAC or OPAL lie too far outside the opacity matrices, it
                is a fatal error and the program halts (normally, such a case
                would simply be signalled by a zero returned value of FEDGE).
                Also, it is a fatal error if you call SET_ALTMIX_MAIN_FILE and
                subsequently use khighz = -2, -3, or -4 (see above), or if any
                filename exceeds 255 characters (or an extension exceeds 80).
  Level = 3 : At this level of error-checking, in addition: if you have read
                the CNO-interpolation opacity files, and you then call the
                subroutine OPAL_X_CNO_FU with a metals-composition array xmet
                with a size nmet other than 19 elements, it is a fatal error
                (you would NOT usually want to use this Level = 3).


*** ASK_ERR_CHECK( LEVEL )  This subroutine returns the error-checking flag
    ----------------------  value LEVEL of as set by SET_ERR_CHECK above.


            ====================================================
            The subroutines that control matrix Z-edge handling:
            ====================================================


*** RESET_Z_LIMITS( vlo, dvlo, vhi, dvhi )  This subroutine can only be called
    --------------------------------------  AFTER a set of opacities has been
  read in (its effects are nullified during opacity input).  WITHOUT affecting
  the stored z-values used for Z-interpolation, calling this subroutine resets
  the range considered to be "interpolation" (which returns FZEDGE = 1.0) and
  the allowed "extrapolation" region (where 0.0 < FZEDGE < 1.0 is returned).
  Negative values (actually, below -1.E-6) mean "leave old value unchanged".
  All these values should be SINGLE PRECISION REAL.
    If  vlo   is non-negative, then this resets  Zlo = vlo .
    If  dvlo  is non-negative, then this resets  Zlo_ex = Zlo - dvlo .
    If  vhi   is non-negative, then this resets  Zhi = vhi .
    If  dvhi  is non-negative, then this resets  Zhi_ex = Zhi + dvhi .
  The values of Zlo and Zhi must not lie outside the range of stored z-values
  used for Z-interpolation, i.e., cases  Zlo < z_low_interpolation_endpoint ,
  Zhi > z_high_interpolation_endpoint , and  Zlo > Zhi  are prohibited.  The
  only constraint on the "extrapolation" region is that  Zlo_ex < Zlo  and
  Zhi_ex > Zhi  (setting  dvlo  and  dvhi  to zero allows extrapolation by
  up to delta-Z = 1.E-6).  Note that FZEDGE is positive (and the opacity is
  calculated) only for the range  Zlo_ex < Z < Zhi_ex .
     NOTE that if  Zlo  and/or  Zhi  is set inside the range covered by the
  stored z-values, the value of FZEDGE will be less than unity for Z outside
  the range  [Zlo,Zhi] , but the actual calculation of opacity values will
  continue to use interpolation (not extrapolation) as long as Z lies inside
  the range of stored z-values; however, for  FZEDGE = 0.0 , the opacity will
  NOT be calculated (even for Z still within the range of stored z-values).


*** ASK_Z_LIMITS( nzmax, zmin, zmax )  This subroutine returns the values of
    ---------------------------------  the hard-wired Z-interpolation limits,
  allowing the user to check what these limiting values actually are.
    INTEGER variable  nzmax  returns NZ: max number of interpolation Z-values.
    SINGLE-PRECISION REAL variable  zmin  returns 0.0 (the lowest allowed Z).
    SINGLE-PRECISION REAL variable  zmax  returns 0.1 (the highest allowed Z).


*** ASK_Z_USE( nzuse, zlo, zmid, zhi, zloex, zhiex )  This subroutine returns
    ------------------------------------------------  the current values of
  the variables controlling Z-interpolation, allowing the user to check what
  the values actually are.
    INTEGER variable  nzuse  returns the number of stored z-values (that will
      be used for Z-interpolation); if no opacity files have been read in yet,
      then  nzuse = 0  is returned (and the other five variables will return
      meaningless values).
    SINGLE-PRECISION REAL variable  zlo  returns the boundary  Zlo  below
      which a Z value is considered to require extrapolation; note that  Zlo
      may lie above the lowest stored z-value, but not below it.
    SINGLE-PRECISION REAL variable  zmid  returns the "typical" Z-value (which
      has no real significance after the opacities have been read in).
    SINGLE-PRECISION REAL variable  zhi  returns the boundary  Zhi  above
      which a Z value is considered to require extrapolation; note that  Zhi
      may lie below the highest stored z-value, but not above it.
    SINGLE-PRECISION REAL variable  zloex  returns the boundary  Zlo_ex  at or
      below which Z-extrapolation is considered too extreme to be carried out.
    SINGLE-PRECISION REAL variable  zhiex  returns the boundary  Zhi_ex  at or
      above which Z-extrapolation is considered too extreme to be carried out.


*** ASK_Z_ARRAY( kzstart, karraystart, Zarray, Narray )  This subroutine will
    ---------------------------------------------------  return (some of) the
  stored z-values (that are used for Z-interpolation), in the user-supplied
  array variable  Zarray  (the other inputs to ASK_Z_ARRAY must have values
  supplied by the user, and may be constant integers).
    kzstart      INTEGER: index of the first stored z-value to be returned.
    karraystart  INTEGER: the index in the user-supplied array Zarray where
                   the first returned z-value will be placed.
    Zarray       SINGLE-PRECISION REAL ARRAY: is where the stored z-values are
                   returned; the array positions  Zarray(karraystart)  through
                   Zarray( min{ Narray , karraystart + nzuse - kzstart } )
                   will contain the stored z-values  kzstart  through  nzuse
                   (where  nzuse  is the total number of stored z-values); any
                   subsequent elements of  Zarray  (up to element  Narray )
                   will be filled with values of -1.0 (note that in no case
                   will elements beyond  Narray  be overwritten).
    Narray       INTEGER: the size of the user-supplied array  Zarray , i.e.,
                   the array is specified as  "dimension Zarray(Narray)" .


         ======================================================
           The subroutines that control matrix T,R-edge handling:
           ======================================================


*** SET_LOGT6_LIMITS( VLO, DVLO, VHI, DVHI )  These subroutines can be called
    ----------------------------------------  at ANY TIME, and their effects
*** SET_LOGR_LIMITS( VLO, DVLO, VHI, DVHI )   will last until they are called
    ---------------------------------------   again; they are used to set (or
  reset) the LogT6 and LogR boundaries.  These input boundaries  VLO  and  VHI
  must not lie outside the matrix edges, and extrapolation is never allowed
  more than one grid-spacing beyond the edge of the matrix.  All these input
  values should be SINGLE-PRECISION REAL, given in terms of  log10(T6)  and
  log10(R) = log10(rho/T6**3); values of -90.0 or less mean "leave the present
  values unchanged", and are ignored.
    If  VLO > -90.0 , then (for subroutine SET_LOGT6_LIMITS) it is used to set
  the lower boundary  LogT6_lo  (minimum -2.25: logT=3.75), or (for subroutine
  SET_LOGR_LIMITS), to set the lower boundary  LogR_lo  (minimum -8.0).
    If  VHI > -90.0 , then it is used to set the upper boundary  LogT6_hi
  (maximum +2.70: logT=8.70) or the upper boundary  LogR_hi  (maximum +1.0).
    If  DVLO  is non-negative, it is used to set the amount of "extrapolation"
  dLogT6 or dLogR allowed beyond the lower boundary, except that the extreme
  values  LogT6_lo - dLogT6  and  LogR_lo - dLogR  are not allowed to lie more
  then one grid spacing beyond the matrix edge; if  -90.0 < DVLO < 0.0 , then
  the amount of extrapolation is set to its default (namely, dLogT6 = 0.05 or
  dLogR = 0.5); if  DVLO < -90.0 , then it is ignored.
    If  DVHI  is non-negative, it is used to set the amount of "extrapolation"
  dLogT6 or dLogR allowed beyond the upper boundary, except that the extreme
  values  LogT6_hi + dLogT6  and  LogR_hi + dLogR  are not allowed to lie more
  then one grid spacing beyond the matrix edge; if  -90.0 < DVHI < 0.0 , then
  the amount of extrapolation is set to its default (namely, dLogT6 = 0.20 or
  dLogR = 0.5); if  DVHI < -90.0 , then it is ignored.
    NOTE that even if the boundaries are set inside the matrix, the opacity
  calculation continues to use all available matrix entries: interpolation is
  still used (not extrapolation) as long as T6 and R lie inside the edge of
  matrix.  The boundaries and "extrapolation" distances are used to obtain the
  value of FTREDGE to return, and whenever  FTREDGE = 0.0  the opacity is NOT
  calculated (even if T6 and R lie inside the matrix edges).


*** ASK_LOGT6_LIMITS( VLO, DVLO, VHI, DVHI )  These subroutines can be called
    ----------------------------------------  at any time; they return the
*** ASK_LOGR_LIMITS( VLO, DVLO, VHI, DVHI )   current values of the lower and
    ---------------------------------------   upper LogT6 or LogR boundaries
  and the corresponding allowed amounts of "extrapolation" dLogT6 or dLogR
  (SINGLE-PRECISION REAL variables must be supplied to hold returned values).

  NOTE ALSO that the OPAL arrays have a "cut-out" region where opacity values
  are not available at high T6,R values; one grid-spacing of extrapolation is
  allowed into this "cut-out" region.  The boundary of this "cut-out" lies
  roughly at LogRho = 4 for 7.0 < logT < 7.5, and at somewhat higher densities
  for log T > 7.5 (up to LogRho = 6 at logT = 8.7, the high-T matrix edge).

  NOTE ALSO that the X=0.0 and X=0.03 matrices have small "cut-outs" at low T6
  and small R.  As noted by Rogers and Iglesias, "as a result of the mixing
  procedure used to calculate the OPAL opacity data, a few X=0.0 and X=0.03
  low T - small R table values fell outside the range of T and R accessible
  from the X=0.35 data directly calculated for this purpose.  These T-R
  locations are filled in with 9.999 (or for diagnostic purposes in some cases
  larger values)."  In the present program, these regions are treated as a
  "cut-out" in the opacity tables (similar to the high T - large R corner),
  and one grid spacing of extrapolation is allowed into them, as at any other
  edge.  For X > 0.1 they have no effect; for 0.03 < X < 0.1, the corner
  ( -8.0 < logR < -7.5 , 3.70 < logT < 3.95 ) [i.e., T6 < 0.008912509] is
  extrapolated; and for X < 0.03, a ragged part of the region ( logR < -4.5 ,
  logT < 4.0 ) [i.e., T6 < 0.01] is considered to be outside the opacity grid
  (i.e., the opacity is not calculated, and FTREDGE = 0.0 is returned).
  Presumably very few users will have applications that take them into these
  low T - small R regions at low hydrogen abundances X; in any case, they are
  at temperatures where molecular opacities may become non-negligible.


           ======================================================
           The subroutines that control how smoothing is handled:
           ======================================================


*** SET_SMOOTH( initsmooth, lowCOsmooth, interpCOsmooth )  This subroutine
    ------------------------------------------------------  allows the user to
  control how and whether the opacity smoothing is carried out when the OPAL
  opacities are read in, and which subroutine is used to interpolate in C and
  O when OPAC or OPAL is called.  The smoothing and its effects are discussed
  in more detail further below.
  --- IT IS RECOMMENDED THAT THE DEFAULT SMOOTHING VALUES NOT BE CHANGED.

    initsmooth      INTEGER: if  initsmooth = 2  (the default), then the OPAL
                      opacities are smoothed by the subroutine OPALTAB when
                      they are read in, in order to remove random numerical
                      errors; if  initsmooth = 0 , then this initial smoothing
                      will not be carried out.  A value  initsmooth = 1  means
                      that opacities used for CNO-interpolation opacity shifts
                      will not be smoothed.  A value   initsmooth < 0  means
                      "do not change the current initial-smoothing setting".
    lowCOsmooth     INTEGER: if  lowCOsmooth = 1  (the default), then the OPAL
                      opacities for the three mixes having max{C,O} = 0.01 may
                      be smoothed in the CO-direction when they are read in;
                      this is only done at (T6,R) points where opacity-changes
                      between mixes with C,O = 0.0, 0.03, 0.1 are monotonic
                      but the opacity at C,O = 0.01 does not fit the trend;
                      the resulting adjustments are small, and only occur at a
                      small minority of (T6,R) points.  If  lowCOsmooth = 0 ,
                      then this initial CO-direction smoothing is not carried
                      out.  A value  lowCOsmooth < 0  means "do not change the
                      current initial-CO-smoothing setting".
    interpCOsmooth  INTEGER: if  interpCOsmooth = 1  (the default), then the
                      subroutine COINTSMO is called by OPAC or OPAL in order
                      to interpolate in C and/or O.  If  interpCOsmooth = 0 ,
                      then the older subroutine COINTERP is used instead; this
                      yields less smooth interpolation, and it has been less
                      thoroughly tested.  A value  interpCOsmooth < 0  means
                      "do not change the current CO-interpolation setting".


*** ASK_SMOOTH( initsmooth, lowCOsmooth, interpCOsmooth )  This subroutine
    ------------------------------------------------------  returns the current
  smoothing settings described above (INTEGER variables must be supplied to
  hold the returned values), allowing the user to check how smoothing is being
  handled.


******************************************************************************



    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    Details of how the opacity interpolation is performed by OPAC or OPAL
    +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

In general, a 6-variable interpolation of log10(Kappa) is performed, using the arguments of OPAC or OPAL, on a subgrid of the stored opacity matrices. In general, 4 stored values ("@" in the diagram at upper right) are used along each interpolation direction. A quadratic fit is performed for each of the 2 sets of 3 adjacent stored values ("1" and "2" at upper right), and then linear interpolation between these overlapping quadratics is used to obtain smoothed results. [For a value near (or beyond) the edge of the matrix, as in the diagrams at lower right, a single quadratic is used.] This procedure produces results that are similar to bicubic spline interpolation, but requires storage of only local information.

```
 1:|-----|--v--|
   @     @  *  @     @
         |--^--|-----|:2

   @  *  @     @     @
 1:|--^--|-----|

*  @     @     @     @
^--|-----|-----|:1
```

--- FIRST, unless excess C = O = 0.0, for each (Z_i , X_j , T6_k , R_n) grid value that will be needed, an interpolation is performed in the excess C and O values ("exC" and "exO" in the arguments to OPAC or OPAL). The actual C’ and O’ values used at each (Z_i , X_j) gridpoint are adjusted by a factor cmod = (1 - X_i - Z_j) / (1 - X - Z) , i.e., C’ and O’ are set to be proportional to the maximum possible value for their (Z_i , X_j) values, so as to avoid out-of-range C’ and O’ values (note that X is "xh", Z is "z" in the arguments to the subroutines OPAC or OPAL described above). The above formula for cmod can lead to problems when (1 - X - Z) is small, therefore: as X is increased from 0.7 to 0.8, cmod switches over smoothly to from the above formula to a constant value of unity (note that since the largest grid {X_i,Z_j} values are X=0.7 and Z=0.1 respectively, at X > 0.8 a value of cmod = 1.0 will never yield X_i + Z_j + C’ + O’ > 1).

The 2-D bi-quadratic interpolation of log(Kappa) in log(C’+Z_i+0.001) and log(O’+Z_i+0.001) is performed by the subroutine COINTSMO (or by the older subroutine COINTERP, if you so choose: see description of subroutine SET_SMOOTH above). The function QCHK is used to evaluate the quadratic: it checks whether 2 of the 3 grid-points are excessively close together (as may happen near C + O = 1 - Z - X for some values of Z) and, if so, uses more nearly liner interpolation to avoid amplifying small errors in the stored opacity values. For the special case where C or O is slightly negative (slight depletion in C or O), the function QCHK does a linear extrapolation using a combination of the lowest three C or O gridpoints. If C and/or O is zero (to within an accuracy of 1.E-6), then interpolation in that direction is not necessary, and is not performed (unless the user has specified that the old subroutine COINTERP should be used).

--- SECOND, unless Z is within 1.E-6 of a stored z-value (or Z < 1.E-8, if the stored value is 0.0), for each (X_j , T6_k , R_n) grid value that will be needed, an interpolation is performed in log(Z+0.001). If there are only 2 stored z-values (numz = 2), linear interpolation is used; for numz = 3, a quadratic is used, while two overlapping quadratics are used for numz > 3 (unless Z is near the end of the range of stored z-values). The subroutine QZLOG4INT is called to perform this Z-interpolation. Since numerical errors in the stored opacities, or in the CO-interpolation, may be comparable to the opacity differences between adjacent stored z-values, the opacity at Z is not allowed to lie outside the range of the two opacities at the stored z-values bracketting it. (Note that, when opacities are read in for values of Z different from one of those available in the OPAL opacity files, the same type of interpolation with the same constraint is performed by the subroutines READZEXCO, READEXCO, or READCO.)

--- THIRD, a two-variable interpolation in performed in the temperature and density variables T6 and R (note slt = log10(T6) and slr = log10(R) in the input to OPAL); the 2-D quadratic interpolation in log10(T6) and log10(R) uses two overlapping quadratics in each direction, unless T6 or R is within one grid spacing of an edge of the table (in which case a single quadratic is used in the relevant direction). NOTE that the high-T,RHO "cutout" has been filled in (by subroutine REVISE_HITR_FOR_INITSMOOTH) with "reasonable" values, at the time when the opacities were read in; for T or RHO above the actual upper edges of the matrix, switch from extrapolation with the slope of the quadratic at the edge of the matrix to 2-point linear extrapolation one grid-spacing beyone the edge of the matrix.

--- FOURTH, unless X is within 1.E-6 of one of the tabulated X-values (X is the input variable xh in OPAC or OPAL), log(Kappa) is interpolated quadratically in log(X+Xdel) , where Xdel = 0.03 is generally used (for 0.03 < X < 0.35, two overlapping quadratics are used). NOTE that pre-1997 versions of the opacity interpolation programs used a value of Xdel = 0.005, which led to non-monotonic behavior of the opacity as a function of X for small X values: for temperatures logT > 5.0 [T6 > 0.1], the interpolated opacity first dropped slightly as X was increased from 0.0 to about 0.005, then increased monotonically thereafter (at least up to X = 0.1). This spurious dip in the opacity for small X values was small (delta log(Kappa) of order 0.03), but it seemed worth getting rid of this dip by setting Xdel to 0.03, in order to obtain qualitatively correct behavior of the opacity for X close to zero. However, at low temperatures (i.e., for logT < 4), the X=0.0 opacities are very small with respect to the X=0.03 and X=0.1 opacities, and a smaller value of Xdel works better near X=0.0. Although such low X values are unlikely to be encountered at such low temperatures, provision was made in the program to reduce the value of Xdel used in such cases to a value that works better (down to a minimum value of 0.001); this was done ONLY for the quadratic that uses opacities at X = 0.0, 0.03, 0.1 (note that for the higher X-values, the value of Xdel used is irrelevant).

--- FIFTH, unless X is within 1.E-6 of one of the tabulated X-values or X < 0.03 or C+O > 0.3 or the "accurate-X" feature was turned off (see the subroutine SET_XHI described above): for the X-values available in 'GN93hz' but not 'Gz???.x??', Z-interpolation and (T6,R)-interpolation is performed in delta-logKappa values, which are then interpolated in X to give opacity corrections. Improvements are small for X < 0.76, but large for X > 0.76 .

--- SIXTH, if (and only if) CNO- and/or user-interpolation is enabled and the corresponding CNO-interconverted opacity-files (or user-specified files) were available to read in: the delta-logKappa values corresponding to the interconversion of the CNO elements (and/or the user-specified composition shift) are multiplied by the relevant factors FCN, FCON, FCNONE, FU and then interpolated in Z, (T6,R), and X in order to give the corresponding opacity corrections.

NOTE: for the Ferguson et al. (2005) molecular opacity interpolation, MODIFIED quadratics are used for logT < 3.48, where sudden jumps in the opacity can occur between one grid-point and the next in any of the Z-, X-, R-, and/or T-directions. For a large jump, the original quadratics would lead to spurious wiggles on each side of the jump, with opacity errors of order a factor of 2 (occasionally more than an order of magnitude). The MODIFIED 3-pt quadratics switch over (smoothly), for a large jump, to linear interpolation in the flat segment, with a quadratic in the "jump" segment whose slope matches that of the flat segment where the two meet at the middle of the three points (two adjacent such modified 3-pt quadratics are still overlapped in general to get the final interpolated value). This gets rid of almost all the spurious wiggles next to opacity jumps. EXAMPLE:

```
Original quadratics:            * *
                               *   *                           * *
  combine              4      4     5   4 * * 5 * 6   -->     4   * 5 * * 6
                       *     *                               *
                       *    *                               *  with spurious
                       *   *                                *  wiggles in
                      *    *                                *  between pts
                     *     *                               *   2 and 3 and
1 * 2 * * 3   2     3      3                  1 * * 2 *   3    between pts
               *   *                                   * *     4 and 5
                * *

Modified quadratics:
                        4       4 * * 5   4 * * 5 * 6    -->     4 * * 5 * 6
  combine               *     *                                 *
                        *    *                                 *
                        *   *                                  *
                       *    *                                  *
                      *     *                                 *
1 * 2 * * 3   2 * * 3       3                      1 * 2 * * 3
```

The modified quadratic routines QUADSL, QDERSL, and QCHKSL are used only for Ferguson et al. 2005 opacity interpolation at logT < 3.48; they are similar to the functions QUAD, QDER, and QCHK [except that QCHKSL assumes that it is always the case that x1 < x2 and that x2 and x3 are the only pair of points that may coincide (or almost coincide)].

Details of the makeup of Z (relative metal abundances), for various mixes

Note that the total metallicity Z for the OPAL mixes includes the following 19 elements: {C,N,O,Ne,Na,Mg,Al,Si,P,S,Cl,Ar,K,Ca,Ti,Cr,Mn,Fe,Ni}. Their relative abundances (by mass fraction and/or number fraction) for any mix can be obtained by calling the subroutine ASK_OPAL_Z_MIX, while their atomic weights (and their atomic numbers, i.e., nuclear charges) can be obtained by calling the subroutine ASK_OPAL_MIX_WT (see descriptions further above).

The makeup of Z in the files 'GN93hz', 'Alrd96a2', 'C95hz', and 'W95hz' is shown below, along with the maximum, mean, and spread of opacity differences relative to 'GN93hz' for Z = 0.1 (where the opacity shifts are largest), for T6 > 0.01 (logT > 4), for each X-value. Note that [i/Fe] gives the log of the enhancement of element i relative to Fe, where the solar reference is the 'GN93hz' mix; note that for i = Fe, [i/Fe] = 0.0 by definition. Newer "GS98" and "AGS04" mixes are given further below. NOTE THAT THE CO-RICH OPACITY FILES 'Gz???.x??' HAVE THE SAME COMPOSITION AS THE FILE 'GN93hz'.

NOTE: for this GN93hz mix, Z/X = 0.02448:

	'Gx..z...' and 'GN93hz'		'Alrd96a2'			'C95hz'			'W95hz'		
	[O/Fe] = 0.0		[O/Fe] = 0.3			[O/Fe] = 0.4			[O/Fe] = 0.5		
i	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z	[i/Fe]	Ni/Nz	Xi/Z	[i/Fe]	Ni/Nz	Xi/Z	[i/Fe]
C	.245518	.173285	.147909	.102693	0.0	.131157	.091924	0.0	.108211	.076451	0.0
N	.064578	.053152	.038904	.031499	0.0	.034498	.028196	0.0	.028462	.023450	0.0
O	.512966	.482273	.616594	.570253	.30	.688325	.642620	.40	.714945	.672836	.50
Ne	.083210	.098668	.100010	.116656	.30	.044451	.052341	0.0	.071502	.084869	.29
Na	.001479	.001999	.001778	.002363	.30	.000790	.001060	0.0	.000652	.000882	0.0
Mg	.026308	.037573	.031622	.044428	.30	.035301	.050066	.40	.029125	.041639	.40
Al	.002042	.003238	.000617	.000962	−.3	.001091	.001718	0.0	.000900	.001428	0.0
Si	.024552	.040520	.029512	.047912	.30	.032945	.053992	.40	.021591	.035669	.30
P	.000195	.000355	.000234	.000420	.30	.000104	.000188	0.0	.000086	.000157	0.0
S	.011222	.021142	.013490	.024999	.30	.015059	.028172	.40	.010575	.019942	.33
Cl	.000219	.000456	.000263	.000539	.30	.000117	.000242	0.0	.000096	.000201	0.0
Ar	.002291	.005379	.002754	.006360	.30	.001224	.002853	0.0	.001010	.002373	0.0
K	.000091	.000210	.000055	.000124	0.0	.000122	.000279	.40	.000040	.000092	0.0
Ca	.001586	.003734	.001906	.004415	.30	.002127	.004975	.40	.002210	.005209	.50
Ti	.000075	.000211	.000089	.000245	.29	.000099	.000275	.39	.000137	.000387	.62
Cr	.000329	.001005	.000198	.000595	0.0	.000176	.000533	0.0	.000145	.000443	0.0
Mn	.000170	.000548	.000072	.00023	−.15	.000036	.000116	−.4	.000075	.000242	0.0
Fe	.021877	.071794	.013177	.042538	0.0	.011687	.038085	0.0	.009642	.031675	0.0
Ni	.001293	.004459	.000816	.002769	.02	.000691	.002365	0.0	.000595	.002056	.02

```
 - ------- ------- ------- ------- --- ------- ------- --- ------- ------- ---
 h>.093729 .192623 .096583 .178899     .101569 .184919     .076879 .142395
================== =================== =================== ===================
 where  h>  is the sum of everything heavier than Ne

 opacity-shifts        'Alrd96a2'          'C95hz'             'W95hz'
  dLogKappa        ------------------- ------------------- -------------------
 for T6 > .01          [O/Fe] = 0.3        [O/Fe] = 0.4        [O/Fe] = 0.5
 for Z = 0.1,      =================== =================== ===================
  relative           max   mean  sigma   max   mean  sigma   max   mean  sigma
 to 'GN93hz'       ------ ------ ----- ------ ------ ----- ------ ------ -----
              X=0: -.1512 -.0270 .0321 -.1844 -.0351 .0371 -.2669 -.0537 .0514
            X=.03: -.1457 -.0258 .0303 -.1835 -.0343 .0364 -.2270 -.0514 .0487
            X=.10: -.1464 -.0249 .0297 -.1849 -.0334 .0361 -.2286 -.0498 .0477
            X=.35: -.1490 -.0236 .0292 -.1886 -.0321 .0359 -.2334 -.0474 .0471
            X=.70: -.1539 -.0227 .0297 -.1952 -.0311 .0367 -.2416 -.0458 .0480

------------------------------------------------------------------------------
 NOTE: if you are using 'Alrd96a2' having [O/Fe] = 0.3 (khighz = 2), then your
        mix Al abundance will go negative if you extrapolate  [O/Fe] > 0.476 ;
        the Mn abundance will go negative if you extrapolate  [O/Fe] > 0.644 .
  --- If you are using 'C95hz' having [O/Fe] = 0.4 (khighz = 3), then your mix
        Mn abundance will go negative if you extrapolate  [O/Fe] > 0.546 .
  --- If you are using 'W95hz' having [O/Fe] = 0.5 (khighz = 4), then your mix
        Ti abundance will go negative if you extrapolate  [O/Fe] < -0.501 .

  By default, the CNO-interpolation files for the "GN93" mix  are  'GN93hz' ,
   'GN93hz.CtoN' , 'GN93hz.COtoN' , 'GN93hz.CNOtoNe' , and 'GN93hz.user' ;
  for other mixes, the default filenames are obtained by appending  '.CtoN' ,
   '.COtoN' , '.CNOtoNe', and '.user' to the main mix filename ('GS98hz' or
  'AGS04hz').  If such non-GN93 files are not found, CNO-interpolation opacity
  differences are obtained from the "GN93" files instead (if these are found).
------------------------------------------------------------------------------



  Opacities for the "GS98" solar/meteoritic abundances (N. Grevesse & A.J.
  Sauval 1998, Space Sci. Rev. 85, 161) are contained in the file  'GS98hz' ;
  three other files were created with opacities for [O/Fe] enhancements (and
  alpha-element enhancements) RELATIVE TO THE "GS98" MIX, patterned after the
  corresponding three cases above.  These files and compositions are:

  NOTE: for this GS98hz mix, Z/X = 0.02300:

        GS98hz    GS98hz_OFe.3_Alrd96a2  GS98hz_OFe.4_C95    GS98hz_OFe.5_W95
   --------------- ------------------- ------------------- -------------------
     [O/Fe] = 0.0      [O/Fe] = 0.3        [O/Fe] = 0.4        [O/Fe] = 0.5
   =============== =================== =================== ===================
 i  Ni/Nz   Xi/Z    Ni/Nz  Xi/Z [i/Fe]  Ni/Nz  Xi/Z [i/Fe]  Ni/Nz  Xi/Z [i/Fe]
 - ------- ------- ------- ------- --- ------- ------- --- ------- ------- ---
 C .245825 .171836 .148069 .101930 0.0 .131883 .091638 0.0 .108877 .076359 0.0
 N .061748 .050335 .037193 .029858 0.0 .033128 .026843 0.0 .027349 .022368 0.0
 O .501922 .467356 .603216 .553139 .30 .676395 .626052 .40 .702986 .656744 .50
 Ne.089265 .104831 .107280 .124072 .30 .047890 .055905 0.0 .077089 .090832 .29
 Na.001562 .002090 .001877 .002473 .30 .000838 .001114 0.0 .000692 .000929 0.0
 Mg.028224 .039924 .033921 .047252 .30 .038036 .053480 .40 .031401 .044563 .40
 Al.002294 .003603 .000693 .001071 -.3 .001231 .001921 0.0 .001016 .001601 0.0
 Si.026954 .044057 .032394 .052144 .30 .036324 .059017 .40 .023820 .039063 .30
 P .000235 .000423 .000282 .000501 .30 .000126 .000226 0.0 .000104 .000188 0.0
 S .012602 .023513 .015145 .027829 .30 .016982 .031497 .40 .011933 .022339 .33
 Cl.000141 .000292 .000170 .000346 .30 .000076 .000156 0.0 .000063 .000130 0.0
 Ar.001865 .004335 .002241 .005131 .30 .001000 .002312 0.0 .000826 .001927 0.0
 K .000100 .000228 .000060 .000135 0.0 .000135 .000305 .40 .000044 .000101 0.0
 Ca.001670 .003896 .002007 .004611 .30 .002251 .005218 .40 .002339 .005474 .50
 Ti.000070 .000195 .000082 .000226 .29 .000092 .000256 .39 .000129 .000362 .62
 Cr.000369 .001117 .000222 .000663 0.0 .000198 .000596 0.0 .000163 .000496 0.0
 Mn.000244 .000779 .000104 .000327-.15 .000052 .000165 -.4 .000108 .000346 0.0
 Fe.023517 .076433 .014165 .045339 0.0 .012616 .040761 0.0 .010416 .033965 0.0
 Ni.001393 .004757 .000878 .002955 .02 .000747 .002537 0.0 .000646 .002214 .02
 - ------- ------- ------- ------- --- ------- ------- --- ------- ------- ---
 h>.101240 .205642 .104241 .191003     .110704 .199561     .083700 .153698
================== =================== =================== ===================
 where  h>  is the sum of everything heavier than Ne


  Opacities can be read in for the newer "AGS04" solar/meteoritic abundances
  (M. Asplund, N. Grevesse, & A.J. Sauval 2004, astro-ph/0410214 [v2];
  ALSO:  M. Asplund, N. Grevesse, & A.J. Sauval 2005, Cosmic Abundances as
    Records of Stellar Evolution and Nucleosynthesis [eds. F.N. Bush & T.G.
    Barnes], ASP Conf. Series, Vol. 336, p. 25;
  ALSO:  M. Asplund, N. Grevesse, & A.J. Sauval 2006, Nucl. Phys. A 777, 1).

  NOTE: for this AGS04hz mix, Z/X = 0.016555973

      AGS04hz    AGS04hz_OFe.3_Alrd96a2  AGS04hz_OFe.4_C95   AGS04hz_OFe.5_W95
   --------------- ------------------- ------------------- -------------------
     [O/Fe] = 0.0      [O/Fe] = 0.3        [O/Fe] = 0.4        [O/Fe] = 0.5
   =============== =================== =================== ===================
 i  Ni/Nz   Xi/Z    Ni/Nz  Xi/Z [i/Fe]  Ni/Nz  Xi/Z [i/Fe]  Ni/Nz  Xi/Z [i/Fe]
 - ------- ------- ------- ------- --- ------- ------- --- ------- ------- ---
 C .257854 .176688 .157301 .106433 0.0 .138833 .094547 0.0 .116312 .080363 0.0
 N .063296 .050578 .038613 .030467 0.0 .034080 .027065 0.0 .028551 .023004 0.0
 O .480145 .438260 .584425 .526743 .30 .649367 .589077 .40 .684888 .630346 .50
 Ne.072700 .083693 .088481 .100581 .30 .039143 .044785 0.0 .063935 .074215 .29
 Na.001956 .002565 .002381 .003084 .30 .001053 .001373 0.0 .000882 .001166 0.0
 Mg.035594 .049354 .043324 .059318 .30 .048138 .066338 .40 .040329 .056385 .40
 Al.002827 .004352 .000865 .001315 -.3 .001522 .002328 0.0 .001275 .001979 0.0
 Si.033992 .054464 .041375 .065461 .30 .045972 .073207 .40 .030593 .049426 .30
 P .000264 .000466 .000321 .000560 .30 .000142 .000249 0.0 .000119 .000212 0.0
 S .015184 .027771 .018483 .033381 .30 .020536 .037330 .40 .014644 .027007 .33
 Cl.000178 .000361 .000216 .000431 .30 .000096 .000193 0.0 .000080 .000163 0.0
 Ar.001590 .003623 .001935 .004355 .30 .000856 .001939 0.0 .000717 .001648 0.0
 K .000121 .000269 .000074 .000163 0.0 .000163 .000361 .40 .000054 .000121 0.0
 Ca.002050 .004686 .002495 .005633 .30 .002771 .006297 .40 .002924 .006742 .50
 Ti.000082 .000223 .000099 .000267 .29 .000109 .000296 .39 .000153 .000422 .62
 Cr.000458 .001358 .000279 .000817 0.0 .000247 .000728 0.0 .000207 .000619 0.0
 Mn.000310 .000972 .000133 .000412-.15 .000066 .000206 -.4 .000140 .000442 0.0
 Fe.029696 .094613 .018112 .056981 0.0 .015989 .050629 0.0 .013395 .043032 0.0
 Ni.001703 .005704 .001088 .003598 .02 .000917 .003052 0.0 .000802 .002708 .02
 - ------- ------- ------- ------- --- ------- ------- --- ------- ------- ---
 h>.126005 .250781 .131180 .235776     .138577 .244526     .106314 .192072
================== =================== =================== ===================
 where  h>  is the sum of everything heavier than Ne

  Unlike earlier papers, "AGS04" above gives meteoritic H, He, C, N, O, Ne, Ar
  abundances.  There are almost no noble gases (He, Ne, Ar ...), less C, N, O
  (relative to Si or Fe) than in the solar mix, and almost no hydrogen.  [For
  other elements, meteoritic abundances were used in the above "AGS04" mix,
  instead of the less-accurate solar-surface observations.]  The actual values
  for the meteoritic mix are given below [with many digits, to show the actual
  abundances of He, Ne, Ar; note that Ab_i = log10(N_i/N_H_sun) + 12.0]:

    Complete "AGS04" meteoritic abundances   Meteoritic mix (components of Z)
    ======================================= ==================================
 i   Ab_i        Ni               Xi               Ni/Nz           Xi/Z
 -- ----- ---------------- ---------------- ---------------- ----------------
 H   8.25 .312727500321341 .020944433890450               [Z=.979055556989581]
 He  1.29 .000000034289886 .000000009119969
 C   7.40 .044173943244219 .035255788483962 .064274280534767 .036009997831346
 N   6.25 .003127275003213 .002910629954482 .004550269595690 .002972895596887
 O   8.39 .431684461811492 .458939274347532 .628112551500661 .468757131371265
 Ne -1.06 .000000000153167 .000000000205377 .000000000222863 .000000000209770
 Na  6.27 .003274596717318 .005002390294590 .004764626668792 .005109403913677
 Mg  7.53 .059588954783339 .096237981691629 .086703538675350 .098296752420816
 Al  6.43 .004732763251461 .008485288886437 .006886298360263 .008666810403005
 Si  7.51 .056906999803204 .106202066768971 .082801221741094 .108473994157720
 P   5.40 .000441970109086 .000909645014595 .000643078446095 .000929104592789
 S   7.16 .025419977789240 .054153160875713 .036986754263261 .055311632204228
 Cl  5.23 .000297994997793 .000702016082572 .000433590770473 .000717033959472
 Ar -0.45 .000000000623973 .000000001656328 .000000000907898 .000000001691761
 K   5.06 .000202569633331 .000526280567971 .000294744287793 .000537539023413
 Ca  6.29 .003431964862220 .009140199460691 .004993601570053 .009335731149717
 Ti  4.89 .000137278594489 .000436941664783 .000199744062802 .000446288938011
 Cr  5.63 .000766751174096 .002649167869084 .001115643667846 .002705840184626
 Mn  5.47 .000518980052336 .001894561100742 .000755129993520 .001935090493299
 Fe  7.45 .049714940755353 .184489597541457 .072336581572827 .188436290692971
 Ni  6.19 .002851042029444 .011120564522665 .004148343157951 .011358461165226
 -- ----- ---------------- ---------------- ---------------- ----------------
 h>       .208286785176682 .481949863998228 .303062898146020 .492259974990732
    ======================================= ===================================
  where  h>  is the sum of everything heavier than Ne

------------------------------------------------------------------------------


  Meteoritic mixes as obtained from solar abundances by reducing C,N,O,Ne,Ar
  as per the AGS04 meteoritic abundances, and "antimeteor" Z-mixes that yield
  solar abundances when added to 50% and 20% as much meteoritic Z-material:

              GN93hz.meteor          GN93hz.5antimeteor GN93hz.2antimeteor
    ================================= ================= =================
 i        Ni/Nz            Xi/Z         Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
 -- ---------------- ---------------- -------- -------- -------- --------
 C  .062096387323045 .034055950767625 .3096329 .2428995 .2728055 .2011308
 N  .004396103132102 .002811584920288 .0856148 .0783222 .0735314 .0632201
 O  .606837084080657 .443326132928943 .4801523 .5017450 .4989998 .4900612
 Ne .000000000215357 .000000000198429 .1122968 .1480020 .0955898 .1184016
 Na .005283859474681 .005546683967014 .0001500 .0002252 .0009138 .0012895
 Mg .093987679773822 .104307149947153 .0026495 .0042059 .0162384 .0242262
 Al .007295227181013 .008987793491910 .0002060 .0003631 .0012607 .0020880
 Si .087714215091835 .112486310445815 .0024732 .0045368 .0151550 .0261267
 P  .000696654916553 .000985278857158 .0000197 .0000399 .0001204 .0002289
 S  .040091598026506 .058690042443895 .0011309 .0023680 .0069273 .0136324
 Cl .000782397036934 .001266565623746 .0000219 .0000507 .0001351 .0002939
 Ar .000000000877017 .000000001599745 .0030924 .0080685 .0026323 .0064548
 K  .000325105632923 .000580403334678 .0000097 .0000248 .0000566 .0001359
 Ca .005666126529857 .010369588461345 .0001590 .0004162 .0009783 .0024069
 Ti .000267944210671 .000586040222841 .0000075 .0000235 .0000463 .0001360
 Cr .001175381889649 .002790592585741 .0000330 .0001122 .0002030 .0006479
 Mn .000607340179662 .001523534791034 .0000168 .0000602 .0001046 .0003529
 Fe .078157536632930 .199305017419850 .0022038 .0080385 .0135039 .0462918
 Ni .004619357794785 .012381327992788 .0001298 .0004978 .0007978 .0028745
 -- ---------------- ---------------- -------- -------- -------- --------
 h> .326670425248838 .519806331184714 .0123032 .0290313 .0590735 .1271863
    ================================= ================= =================
  (NOTE: some abundances would go negative beyond GN93hz.561743antimeteor)

              GS98hz.meteor          GS98hz.5antimeteor GS98hz.2antimeteor
    ================================= ================= =================
 i        Ni/Nz            Xi/Z         Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
 -- ---------------- ---------------- -------- -------- -------- --------
```

```
C  .061833419528987 .033847976603222 .3108195 .2408300 .2734442 .1994336
N  .004377442231746 .002794386850040 .0820144 .0741053 .0703603 .0598431
O  .604274480285562 .440624098408040 .4657653 .4807222 .4865568 .4727025
Ne .000000000214427 .000000000197201 .1207976 .1572465 .1026646 .1257972
Na .005179390157206 .005426796827570 .0002843 .0004216 .0010191 .0014226
Mg .093587138052187 .103667372071604 .0051357 .0080523 .0184132 .0271753
Al .007606607490520 .009353799223261 .0004180 .0007276 .0014971 .0024528
Si .089375982574740 .114401914525715 .0049037 .0088845 .0175839 .0299880
P  .000779229621820 .001099992396139 .0000423 .0000845 .0001529 .0002876
S  .041786601600009 .061056356041069 .0022925 .0047413 .0082210 .0160043
Cl .000467537763442 .000755439983010 .0000264 .0000603 .0000926 .0001993
Ar .000000000873527 .000000001590383 .0025233 .0065025 .0021445 .0052020
K  .000331587065415 .000590861589288 .0000185 .0000466 .0000655 .0001554
Ca .005537504221631 .010115144589889 .0003042 .0007864 .0010898 .0026522
Ti .000232110957854 .000506712319369 .0000127 .0000391 .0000456 .0001327
Cr .001223556258112 .002899506965060 .0000673 .0002257 .0002409 .0007605
Mn .000809072444438 .002025771911462 .0000439 .0001556 .0001588 .0005296
Fe .077979330676706 .198476752244634 .0042777 .0154111 .0153411 .0520243
Ni .004619007981672 .012357115663043 .0002527 .0009569 .0009081 .0032370
-- ---------------- ---------------- -------- -------- -------- --------
h> .329514657739278 .522733537941497 .0206032 .0470960 .0669741 .1422236
   ================================= ================= =================
 (NOTE: some abundances would go negative beyond GS98hz.624814antimeteor)

           AGS04hz.meteor         AGS04hz.5antimeteor AGS04hz.2antimeteor
   ================================= ================= =================
i        Ni/Nz            Xi/Z         Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ---------------- ---------------- -------- -------- -------- --------
C  .064274208429525 .036009888957169 .3303785 .2470271 .2883952 .2048236
N  .004550290959465 .002972903901491 .0853040 .0743805 .0725636 .0600990
O  .628110449385059 .468754671183736 .4247136 .4230127 .4568036 .4321613
Ne .000000000222945 .000000000209847 .0999372 .1255395 .0841701 .1004316
Na .004764660515098 .005109430493010 .0009033 .0012928 .0015125 .0020561
Mg .086704154588141 .098297263764627 .0164453 .0248824 .0275300 .0395653
Al .006886347278212 .008666855488115 .0013066 .0021946 .0021869 .0034890
Si .082801809933137 .108474558443967 .0157052 .0274587 .0262910 .0436619
P  .000643083014308 .000929109426030 .0001216 .0002344 .0002039 .0003734
S  .036987017004729 .055311919937648 .0070150 .0140005 .0117437 .0222628
Cl .000433593850556 .000717037689513 .0000829 .0001830 .0001382 .0002898
Ar .000000000907947 .000000001691848 .0021853 .0054345 .0018405 .0043476
K  .000294746381558 .000537541819713 .0000553 .0001347 .0000931 .0002153
Ca .004993637042920 .009335779714580 .0009463 .0023611 .0015848 .0037560
Ti .000199745481717 .000446291259625 .0000374 .0001114 .0000630 .0001783
Cr .001115651593004 .002705854260520 .0002113 .0006841 .0003540 .0010884
Mn .000755135357710 .001935100559721 .0001434 .0004904 .0002399 .0007794
Fe .072337095427584 .188437270946424 .0137206 .0477009 .0229685 .0758481
Ni .004148372626387 .011358520252416 .0007872 .0028767 .0013175 .0045731
-- ---------------- ---------------- -------- -------- -------- --------
h> .303065051003006 .492262535747757 .0596667 .1300402 .0980675 .2024845
   ================================= ================= =================
 (NOTE: some abundances would go negative beyond AGS04hz.998696antimeteor)

 Since the solar C, N, and O abundances have been reduced more-or-less in
 concert from one abundance paper to the next, it is not unreasonable to
 consider them NOT to be independent, and to vary all of them together by
 their quoted uncertainties (or by double this amount) to get mixes with
 shifted C, N, and O that can still be considered "reasonable" (note that Ne
 and Ar should be shifted by the same amount as O, since it is their ratios
 Ne/O and Ar/O that are measured).  The shifted-CNO mixes would have the same
 meteoritic abundances as "AGS04" above, but different "anti-meteor" mixes:

 Z/X = 0.01809864    AGS04hiCNONe_OFe.5_W95         AGS04hiCNONe.2antimeteor
        AGS04hiCNONe          ||     AGS04hiCNONe.5antimeteor    ||
   ==================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +0.05 .261002 .181349 .117585 .082019 .3333123 .2540186 .2915434 .2104168
N  +0.06 .065561 .053122 .029536 .024025 .0879866 .0781965 .0750329 .0631518
O  +0.05 .486005 .449821 .692380 .643323 .4337738 .4403542 .4639452 .4460345
Ne +0.05 .073584 .085897 .064632 .075741 .1006317 .1288455 .0850084 .1030764
Na  0.00 .001765 .002347 .000795 .001061 .0006621 .0009658 .0012990 .0017945
Mg  0.00 .032110 .045147 .036336 .051288 .0120428 .0185719 .0236341 .0345169
Al  0.00 .002551 .003981 .001149 .001800 .0009568 .0016381 .0018774 .0030438
Si  0.00 .030665 .049822 .027564 .044958 .0115013 .0204957 .0225709 .0380915
P   0.00 .000238 .000427 .000107 .000192 .0000895 .0001759 .0001755 .0003266
S   0.00 .013698 .025404 .013194 .024565 .0051371 .0104500 .0100819 .0194224
Cl  0.00 .000161 .000330 .000072 .000148 .0000607 .0001365 .0001186 .0002526
Ar +0.05 .001609 .003719 .000725 .001682 .0022008 .0055785 .0018591 .0044628
K   0.00 .000109 .000246 .000049 .000111 .0000404 .0001002 .0000799 .0001877
Ca  0.00 .001849 .004287 .002634 .006131 .0006931 .0017626 .0013607 .0032772
Ti  0.00 .000074 .000204 .000138 .000384 .0000273 .0000829 .0000540 .0001555
Cr  0.00 .000413 .001242 .000186 .000562 .0001546 .0005101 .0003038 .0009492
Mn  0.00 .000280 .000889 .000126 .000402 .0001050 .0003659 .0002059 .0006798
Fe  0.00 .026789 .086548 .012069 .039143 .0100475 .0356034 .0197181 .0661701
Ni  0.00 .001537 .005218 .000723 .002465 .0005766 .0021477 .0011312 .0039899
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .113848 .229811 .095867 .174892 .0442956 .0985852 .0844701 .1773205
   ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04hiCNONe.841962amet)

 Z/X = 0.01983278    AGS04vhCNONe_OFe.5_W95          AGS04vhCNONe.2antimeteor
        AGS04vhCNONe          ||     AGS04vhCNONe.5antimeteor    ||
   ==================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +0.10 .263820 .185686 .118732 .083546 .3358885 .2605241 .2943424 .2156212
N  +0.12 .067812 .055659 .030518 .025042 .0906599 .0820020 .0774886 .0661962
O  +0.10 .491254 .460576 .699138 .655310 .4418255 .4564867 .4703199 .4589405
Ne +0.10 .074376 .087948 .065260 .077148 .1012381 .1319220 .0857528 .1055376
Na  0.00 .001590 .002142 .000716 .000964 .0004434 .0006583 .0011044 .0015485
Mg  0.00 .028927 .041200 .032700 .046561 .0080606 .0126514 .0200899 .0297805
Al  0.00 .002298 .003633 .001034 .001634 .0006406 .0011161 .0015959 .0026262
Si  0.00 .027625 .045465 .024806 .040815 .0076973 .0139602 .0191852 .0328631
P   0.00 .000214 .000389 .000096 .000174 .0000594 .0001189 .0001487 .0002810
S   0.00 .012340 .023183 .011874 .022302 .0034384 .0071185 .0085700 .0167572
Cl  0.00 .000145 .000301 .000065 .000135 .0000406 .0000930 .0001007 .0002178
Ar +0.10 .001627 .003808 .000732 .001713 .0022142 .0057120 .0018755 .0045696
K   0.00 .000098 .000225 .000044 .000101 .0000272 .0000687 .0000681 .0001625
Ca  0.00 .001666 .003912 .002370 .005565 .0004637 .0012001 .0011566 .0028272
Ti  0.00 .000066 .000186 .000123 .000345 .0000181 .0000559 .0000458 .0001339
Cr  0.00 .000372 .001133 .000167 .000509 .0001032 .0003466 .0002581 .0008184
Mn  0.00 .000252 .000811 .000114 .000367 .0000702 .0002489 .0001749 .0005862
Fe  0.00 .024134 .078981 .010861 .035534 .0067250 .0242529 .0167609 .0570897
Ni  0.00 .001384 .004762 .000650 .002235 .0003861 .0014637 .0009616 .0034427
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .102738 .210131 .086352 .158954 .0303880 .0690652 .0720963 .1537045
   ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04vhCNONe.714584amet)
```

The components of the metallicity for the CNO−varied mixes are given below. Note that burning C to N increases Z slightly, burning O to N decreases Z slightly, and burning CNO to Ne increases Z significantly; thus, although the mass fraction <Xheavy> of elements heavier than Ne does not change, the ratio <h> = <Xheavy>/Z differs between these CNO−varied mixes:

	’GN93hz’			’GN93hz.CtoN’		’GN93hz.COtoN’		’GN93hz.CNOtoNe’	
	solar CNO			most C --> N		most C,O --> N		all C,N,O --> Ne	
i	Ab_i	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z
C	8.55	.245518	.173285	.000246	.000169	.000246	.000179	.000000	.000000
N	7.97	.064578	.053152	.309850	.247897	.817686	.694328	.000000	.000000
O	8.87	.512966	.482273	.512966	.468789	.005130	.004976	.000000	.000000
Ne	8.08	.083210	.098668	.083210	.095909	.083210	.101793	.906272	.847998
Na	6.33	.001479	.001999	.001479	.001942	.001479	.002061	.001479	.001577
Mg	7.58	.026308	.037573	.026308	.036523	.026308	.038764	.026308	.029650
Al	6.47	.002042	.003238	.002042	.003147	.002042	.003340	.002042	.002555
Si	7.55	.024552	.040520	.024552	.039387	.024552	.041803	.024552	.031975
P	5.45	.000195	.000355	.000195	.000345	.000195	.000366	.000195	.000280
S	7.21	.011222	.021142	.011222	.020550	.011222	.021811	.011222	.016683
Cl	5.50	.000219	.000456	.000219	.000443	.000219	.000471	.000219	.000360
Ar	6.52	.002291	.005379	.002291	.005228	.002291	.005548	.002291	.004244
K	5.12	.000091	.000210	.000091	.000203	.000091	.000216	.000091	.000165
Ca	6.36	.001586	.003734	.001586	.003631	.001586	.003854	.001586	.002948
Ti	5.03	.000075	.000211	.000075	.000205	.000075	.000218	.000075	.000167
Cr	5.68	.000329	.001005	.000329	.000977	.000329	.001037	.000329	.000793
Mn	5.39	.000170	.000548	.000170	.000533	.000170	.000566	.000170	.000433
Fe	7.50	.021877	.071794	.021877	.069787	.021877	.074068	.021877	.056653
Ni	6.25	.001293	.004459	.001293	.004335	.001293	.004601	.001293	.003519
<h>		.093729	.192623	.093729	.187236	.093729	.198724	.093729	.152002

where <h> is the sum of everything heavier than Ne.

For the GS98 mix, the corresponding CNO−varied metallicity components are:

	’GS98hz’			’GS98hz.CtoN’		’GS98hz.COtoN’		’GS98hz.CNOtoNe’	
	solar CNO			most C --> N		most C,O --> N		all C,N,O --> Ne	
i	Ab_i	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z	Ni/Nz	Xi/Z
C	8.52	.245825	.171836	.000246	.000167	.000246	.000177	.000000	.000000
N	7.92	.061748	.050335	.307327	.243574	.804230	.675229	.000000	.000000
O	8.83	.501922	.467356	.501922	.454393	.005019	.004813	.000000	.000000
Ne	8.08	.089265	.104831	.089265	.101924	.089265	.107973	.898760	.836936
Na	6.32	.001562	.002090	.001562	.002032	.001562	.002153	.001562	.001657
Mg	7.58	.028224	.039924	.028224	.038816	.028224	.041120	.028224	.031657
Al	6.49	.002294	.003603	.002294	.003502	.002294	.003710	.002294	.002856
Si	7.56	.026954	.044057	.026954	.042835	.026954	.045377	.026954	.034935
P	5.56	.000235	.000423	.000235	.000412	.000235	.000436	.000235	.000336
S	7.20	.012602	.023513	.012602	.022861	.012602	.024218	.012602	.018645
Cl	5.28	.000141	.000292	.000141	.000283	.000141	.000300	.000141	.000231
Ar	6.40	.001865	.004335	.001865	.004216	.001865	.004466	.001865	.003438
K	5.13	.000100	.000228	.000100	.000221	.000100	.000234	.000100	.000180
Ca	6.35	.001670	.003896	.001670	.003787	.001670	.004012	.001670	.003089
Ti	4.94	.000070	.000195	.000070	.000190	.000070	.000201	.000070	.000155
Cr	5.69	.000369	.001117	.000369	.001086	.000369	.001150	.000369	.000885
Mn	5.53	.000244	.000779	.000244	.000759	.000244	.000804	.000244	.000619
Fe	7.50	.023517	.076433	.023517	.074315	.023517	.078726	.023517	.060608
Ni	6.25	.001393	.004757	.001393	.004627	.001393	.004901	.001393	.003773
<h>		.101240	.205642	.101240	.199942	.101240	.211808	.101240	.163064

where <h> is the sum of everything heavier than Ne.

```
 For the AGS04 mix, the corresponding CNO-varied metallicity components are:

          ’AGS04hz’        ’AGS04hz.CtoN’  ’AGS04hz.COtoN’  ’AGS04hz.CNOtoNe’
    --------------------  ---------------  ---------------  ----------------
          solar CNO         most C --> N    most C,O --> N   all C,N,O --> Ne
    ====================  ===============  ===============  ================
i   Ab_i  Ni/Nz   Xi/Z     Ni/Nz   Xi/Z     Ni/Nz   Xi/Z     Ni/Nz   Xi/Z
--  ---- ------- -------  ------- -------  ------- -------  ------- -------
C   8.39 .257854 .176688  .002579 .001717  .002579 .001812  .000000 .000000
N   7.78 .063296 .050578  .318571 .247373  .793915 .650653  .000000 .000000
O   8.66 .480145 .438260  .480145 .425880  .004801 .004494  .000000 .000000
Ne  7.84 .072700 .083693  .072700 .081329  .072700 .085836  .873995 .800479
Na  6.27 .001956 .002565  .001956 .002493  .001956 .002631  .001956 .002041
Mg  7.53 .035594 .049354  .035594 .047960  .035594 .050618  .035594 .039266
Al  6.43 .002827 .004352  .002827 .004229  .002827 .004463  .002827 .003462
Si  7.51 .033992 .054464  .033992 .052926  .033992 .055859  .033992 .043331
P   5.40 .000264 .000466  .000264 .000453  .000264 .000478  .000264 .000371
S   7.16 .015184 .027771  .015184 .026987  .015184 .028483  .015184 .022095
Cl  5.23 .000178 .000361  .000178 .000350  .000178 .000369  .000178 .000286
Ar  6.18 .001590 .003623  .001590 .003521  .001590 .003716  .001590 .002883
K   5.06 .000121 .000269  .000121 .000262  .000121 .000277  .000121 .000215
Ca  6.29 .002050 .004686  .002050 .004555  .002050 .004807  .002050 .003729
Ti  4.89 .000082 .000223  .000082 .000218  .000082 .000230  .000082 .000178
Cr  5.63 .000458 .001358  .000458 .001320  .000458 .001393  .000458 .001081
Mn  5.47 .000310 .000972  .000310 .000944  .000310 .000996  .000310 .000773
Fe  7.45 .029696 .094613  .029696 .091941  .029696 .097036  .029696 .075273
Ni  6.19 .001703 .005704  .001703 .005542  .001703 .005849  .001703 .004537
--  ---- ------- -------  ------- -------  ------- -------  ------- -------
<h>      .126005 .250781  .126005 .243701  .126005 .257205  .126005 .199521
    ====================  ===============  ===============  ================
where  <h>  is the sum of everything heavier than Ne.

 For the high-C,N,O,Ne version of the "AGS04" mix:

                       ’AGS04hiCNONe.CtoN’            ’AGS04hiCNONe.CNOtoNe’
         ’AGS04hiCNONe’         ||      ’AGS04hiCNONe.COtoN’      ||
    --------------------  ---------------  ---------------  ----------------
          solar CNO         most C --> N    most C,O --> N   all C,N,O --> Ne
    ====================  ===============  ===============  ================
i   Ab_i  Ni/Nz   Xi/Z     Ni/Nz   Xi/Z     Ni/Nz   Xi/Z     Ni/Nz   Xi/Z
--  ---- ------- -------  ------- -------  ------- -------  ------- -------
C   8.44 .261002 .181349  .002610 .001761  .002610 .001861  .000000 .000000
N   7.84 .065561 .053122  .323953 .254885  .805098 .669508  .000000 .000000
O   8.71 .486005 .449821  .486005 .436791  .004860 .004616  .000000 .000000
Ne  7.89 .073584 .085897  .073584 .083408  .073584 .088156  .886152 .818221
Na  6.27 .001765 .002347  .001765 .002279  .001765 .002409  .001765 .001857
Mg  7.53 .032110 .045147  .032110 .043839  .032110 .046335  .032110 .035711
Al  6.43 .002551 .003981  .002551 .003866  .002551 .004086  .002551 .003149
Si  7.51 .030665 .049822  .030665 .048378  .030665 .051132  .030665 .039408
P   5.40 .000238 .000427  .000238 .000414  .000238 .000438  .000238 .000337
S   7.16 .013698 .025404  .013698 .024669  .013698 .026073  .013698 .020095
Cl  5.23 .000161 .000330  .000161 .000321  .000161 .000339  .000161 .000261
Ar  6.23 .001609 .003719  .001609 .003611  .001609 .003816  .001609 .002941
K   5.06 .000109 .000246  .000109 .000239  .000109 .000253  .000109 .000195
Ca  6.29 .001849 .004287  .001849 .004163  .001849 .004400  .001849 .003391
Ti  4.89 .000074 .000204  .000074 .000199  .000074 .000210  .000074 .000162
Cr  5.63 .000413 .001242  .000413 .001206  .000413 .001275  .000413 .000983
Mn  5.47 .000280 .000889  .000280 .000864  .000280 .000913  .000280 .000704
Fe  7.45 .026789 .086548  .026789 .084039  .026789 .088823  .026789 .068457
Ni  6.19 .001537 .005218  .001537 .005068  .001537 .005357  .001537 .004128
--  ---- ------- -------  ------- -------  ------- -------  ------- -------
<h>      .113848 .229811  .113848 .223155  .113848 .235859  .113848 .181779
    ====================  ===============  ===============  ================
where  <h>  is the sum of everything heavier than Ne.

 For the very-high-C,N,O,Ne version of the "AGS04" mix:

                       'AGS04vhCNONe.CtoN'            'AGS04vhCNONe.CNOtoNe'
         'AGS04vhCNONe'         ||      'AGS04vhCNONe.COtoN'      ||
    --------------------  ---------------  ---------------  ----------------
          solar CNO        most C --> N    most C,O --> N   all C,N,O --> Ne
    ====================  ===============  ===============  ================
i   Ab_i  Ni/Nz   Xi/Z     Ni/Nz   Xi/Z     Ni/Nz   Xi/Z     Ni/Nz   Xi/Z
--  ---- ------- -------  ------- -------  ------- -------  ------- -------
C   8.49 .263820 .185686  .002638 .001802  .002638 .001907  .000000 .000000
N   7.90 .067812 .055659  .328994 .262029  .815335 .687252  .000000 .000000
O   8.76 .491254 .460576  .491254 .446925  .004913 .004730  .000000 .000000
Ne  7.94 .074376 .087948  .074376 .085341  .074376 .090318  .897262 .834689
Na  6.27 .001590 .002142  .001590 .002079  .001590 .002200  .001590 .001685
Mg  7.53 .028927 .041200  .028927 .039978  .028927 .042310  .028927 .032412
Al  6.43 .002298 .003633  .002298 .003526  .002298 .003731  .002298 .002858
Si  7.51 .027625 .045465  .027625 .044117  .027625 .046690  .027625 .035768
P   5.40 .000214 .000389  .000214 .000377  .000214 .000399  .000214 .000306
S   7.16 .012340 .023183  .012340 .022496  .012340 .023808  .012340 .018238
Cl  5.23 .000145 .000301  .000145 .000292  .000145 .000309  .000145 .000237
Ar  6.28 .001627 .003808  .001627 .003696  .001627 .003911  .001627 .002996
K   5.06 .000098 .000225  .000098 .000218  .000098 .000231  .000098 .000177
Ca  6.29 .001666 .003912  .001666 .003797  .001666 .004018  .001666 .003078
Ti  4.89 .000066 .000186  .000066 .000180  .000066 .000190  .000066 .000146
Cr  5.63 .000372 .001133  .000372 .001100  .000372 .001164  .000372 .000892
Mn  5.47 .000252 .000811  .000252 .000787  .000252 .000833  .000252 .000638
Fe  7.45 .024134 .078981  .024134 .076640  .024134 .081110  .024134 .062135
Ni  6.19 .001384 .004762  .001384 .004620  .001384 .004889  .001384 .003745
--  ---- ------- -------  ------- -------  ------- -------  ------- -------
<h>      .102738 .210131  .102738 .203903  .102738 .215793  .102738 .165311
    ====================  ===============  ===============  ================
where  <h>  is the sum of everything heavier than Ne.

------------------------------------------------------------------------------


 There are indications that Ne might have an abundance up to 3 times that
 inferred from the observed Solar coronal Ne/O ratios, although there are
 arguments against this too; at the very least, the Ne abundance may be quite
 uncertain, so here are AGS04 mixes with Ne * 1.5, 2.0, 2.5, and 3.0:

 Z/X = 0.01724878      AGS04x15Ne_OFe.5_W95
           AGS04x15Ne        ||          AGS04x15Ne.5amet  AGS04x15Ne.2amet
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  0.000 .248810 .169591 .112709 .077488 .3184504 .2363816 .2781026 .1963072
N  0.000 .061076 .048547 .027666 .022181 .0824077 .0713340 .0700487 .0576618
O  0.000 .463305 .420657 .663672 .607789 .4011116 .3966082 .4371453 .4110377
Ne +.176 .105225 .120496 .092932 .107340 .1449344 .1807440 .1219276 .1445952
Na 0.000 .001887 .002462 .000855 .001125 .0008012 .0011383 .0014303 .0019325
Mg 0.000 .034345 .047372 .039079 .054367 .0145862 .0219094 .0260341 .0371869
Al 0.000 .002728 .004177 .001236 .001909 .0011587 .0019321 .0020679 .0032790
Si 0.000 .032800 .052277 .029646 .047659 .0139299 .0241782 .0248626 .0410375
P  0.000 .000255 .000448 .000116 .000206 .0001083 .0002074 .0001933 .0003518
S  0.000 .014651 .026656 .014190 .026040 .0062221 .0123280 .0111057 .0209248
Cl 0.000 .000172 .000346 .000077 .000156 .0000733 .0001605 .0001305 .0002718
Ar 0.000 .001534 .003478 .000695 .001589 .0021132 .0052170 .0017777 .0041736
K  0.000 .000116 .000258 .000052 .000116 .0000489 .0001182 .0000880 .0002021
Ca 0.000 .001978 .004498 .002833 .006499 .0008394 .0020791 .0014988 .0035304
Ti 0.000 .000079 .000214 .000148 .000406 .0000331 .0000979 .0000595 .0001675
Cr 0.000 .000442 .001303 .000200 .000595 .0001872 .0006016 .0003346 .0010224
```

```
Mn 0.000 .000299 .000933 .000136 .000428 .0001272 .0004319 .0002269 .0007326
Fe 0.000 .028654 .090812 .012980 .041493 .0121689 .0419994 .0217199 .0712869
Ni 0.000 .001644 .005475 .000778 .002614 .0006983 .0025332 .0012460 .0042983
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .121584 .240709 .103021 .185202 .0530959 .1149322 .0927758 .1903981
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04x15Ne.921257amet)

 Z/X = 0.01794158      AGS04x2Ne_OFe.5_W95
           AGS04x2Ne          ||          AGS04x2Ne.5amet   AGS04x2Ne.2amet
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  0.000 .240379 .163043 .109323 .074812 .3072900 .2265596 .2684938 .1884496
N  0.000 .059006 .046672 .026835 .021415 .0796959 .0685215 .0676994 .0554118
O  0.000 .447605 .404412 .643730 .586793 .3790228 .3722407 .4187884 .3915437
Ne +.301 .135546 .154458 .120186 .138176 .1870455 .2316870 .1571847 .1853496
Na 0.000 .001823 .002367 .000829 .001086 .0007056 .0009958 .0013536 .0018185
Mg 0.000 .033182 .045543 .037906 .052491 .0128463 .0191659 .0246373 .0349921
Al 0.000 .002636 .004016 .001199 .001843 .0010207 .0016906 .0019571 .0030858
Si 0.000 .031688 .050258 .028755 .046012 .0122678 .0211497 .0235282 .0386147
P  0.000 .000246 .000430 .000112 .000198 .0000949 .0001804 .0001824 .0003302
S  0.000 .014155 .025626 .013764 .025141 .0054793 .0107830 .0105093 .0196888
Cl 0.000 .000166 .000333 .000075 .000151 .0000648 .0001410 .0001237 .0002562
Ar 0.000 .001482 .003344 .000674 .001534 .0020455 .0050160 .0017190 .0040128
K  0.000 .000112 .000248 .000051 .000114 .0000430 .0001032 .0000832 .0001901
Ca 0.000 .001911 .004324 .002748 .006275 .0007390 .0018181 .0014182 .0033216
Ti 0.000 .000076 .000206 .000143 .000390 .0000292 .0000859 .0000564 .0001579
Cr 0.000 .000427 .001253 .000194 .000575 .0001650 .0005266 .0003167 .0009624
Mn 0.000 .000289 .000897 .000132 .000413 .0001121 .0003779 .0002147 .0006894
Fe 0.000 .027683 .087306 .012590 .040059 .0107174 .0367404 .0205546 .0670797
Ni 0.000 .001588 .005264 .000754 .002522 .0006152 .0022167 .0011793 .0040451
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .117464 .231415 .099926 .178804 .0469458 .1009912 .0878337 .1792453
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04x2Ne.856526amet)

 Z/X = 0.01863439      AGS04x25Ne_OFe.5_W95
           AGS04x25Ne         ||          AGS04x25Ne.5amet  AGS04x25Ne.2amet
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  0.000 .232500 .156981 .106134 .072313 .2968211 .2174666 .2594992 .1811752
N  0.000 .057072 .044937 .026052 .020700 .0771536 .0659190 .0655014 .0533298
O  0.000 .432932 .389378 .624950 .567196 .3583114 .3496897 .4016121 .3735029
Ne +.398 .163879 .185894 .145851 .166953 .2265366 .2788410 .1901794 .2230728
Na 0.000 .001764 .002279 .000805 .001050 .0006160 .0008638 .0012818 .0017129
Mg 0.000 .032094 .043849 .036800 .050737 .0112136 .0166249 .0233291 .0329593
Al 0.000 .002549 .003867 .001164 .001782 .0008914 .0014671 .0018535 .0029070
Si 0.000 .030650 .048389 .027916 .044475 .0107089 .0183462 .0222792 .0363719
P  0.000 .000238 .000414 .000109 .000192 .0000828 .0001564 .0001727 .0003110
S  0.000 .013691 .024674 .013362 .024301 .0047837 .0093550 .0099520 .0185464
Cl 0.000 .000161 .000321 .000073 .000147 .0000569 .0001230 .0001173 .0002418
Ar 0.000 .001434 .003219 .000655 .001484 .0019815 .0048285 .0016635 .0038628
K  0.000 .000109 .000239 .000050 .000111 .0000376 .0000897 .0000789 .0001793
Ca 0.000 .001848 .004164 .002667 .006064 .0006455 .0015781 .0013433 .0031296
Ti 0.000 .000074 .000198 .000140 .000380 .0000253 .0000739 .0000533 .0001483
Cr 0.000 .000413 .001206 .000189 .000557 .0001438 .0004561 .0002998 .0009060
Mn 0.000 .000280 .000863 .000128 .000399 .0000975 .0003269 .0002031 .0006486
Fe 0.000 .026776 .084060 .012223 .038722 .0093558 .0318714 .0194638 .0631845
Ni 0.000 .001536 .005068 .000732 .002437 .0005370 .0019227 .0011166 .0038099
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .113617 .222810 .097013 .172838 .0411773 .0880837 .0832079 .1689193
```

```
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04x25Ne.797451amet)

 Z/X = 0.01932720      AGS04x3Ne_OFe.5_W95
           AGS04x3Ne          ||          AGS04x3Ne.5amet   AGS04x3Ne.2amet
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  0.000 .225122 .151354 .103126 .069977 .2869847 .2090261 .2510645 .1744228
N  0.000 .055261 .043326 .025314 .020031 .0747640 .0635025 .0634395 .0513966
O  0.000 .419197 .375420 .607243 .548880 .3388465 .3287527 .3855011 .3567533
Ne +.477 .190413 .215077 .170059 .193870 .2636471 .3226155 .2211244 .2580924
Na 0.000 .001708 .002198 .000783 .001017 .0005325 .0007423 .0012150 .0016157
Mg 0.000 .031075 .042278 .035756 .049097 .0096809 .0142684 .0221037 .0310741
Al 0.000 .002468 .003728 .001131 .001724 .0007692 .0012586 .0017558 .0027402
Si 0.000 .029677 .046655 .027125 .043039 .0092449 .0157452 .0211086 .0342911
P  0.000 .000230 .000399 .000105 .000184 .0000713 .0001339 .0001635 .0002930
S  0.000 .013256 .023789 .012983 .023515 .0041291 .0080275 .0094286 .0174844
Cl 0.000 .000156 .000309 .000071 .000142 .0000488 .0001050 .0001109 .0002274
Ar 0.000 .001388 .003104 .000636 .001435 .0019220 .0046560 .0016120 .0037248
K  0.000 .000105 .000230 .000048 .000106 .0000321 .0000762 .0000745 .0001685
Ca 0.000 .001789 .004014 .002591 .005867 .0005567 .0013531 .0012723 .0029496
Ti 0.000 .000071 .000191 .000135 .000365 .0000218 .0000634 .0000505 .0001399
Cr 0.000 .000400 .001163 .000183 .000538 .0001242 .0003916 .0002841 .0008544
Mn 0.000 .000271 .000832 .000124 .000385 .0000842 .0002804 .0001924 .0006114
Fe 0.000 .025926 .081047 .011876 .037470 .0080765 .0273519 .0184408 .0595689
Ni 0.000 .001487 .004886 .000711 .002358 .0004635 .0016497 .0010578 .0035915
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .110007 .214823 .094258 .167242 .0357577 .0761032 .0788705 .1593349
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04x3Ne.747866amet)

 Z/X = 0.01887595    AGS04hiCNOx15Ne_OFe.5_W95         AGS04hiCNOx15Ne.2amet
        AGS04hiCNOx15Ne       ||       AGS04hiCNOx15Ne.5amet     ||
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.050 .251740 .173881 .113904 .079026 .3212059 .2428166 .2810428 .2014552
N  +.060 .063234 .050934 .028611 .023148 .0849795 .0749145 .0724070 .0605262
O  +.050 .468761 .431300 .670709 .619856 .4097142 .4125727 .4438537 .4238093
Ne +.226 .106459 .123540 .093912 .109465 .1459095 .1853100 .1231012 .1482480
Na 0.000 .001702 .002250 .000770 .001023 .0005669 .0008203 .0012231 .0016781
Mg 0.000 .030971 .043288 .035199 .049417 .0103179 .0157834 .0222584 .0322861
Al 0.000 .002460 .003817 .001113 .001735 .0008198 .0013921 .0017680 .0028470
Si 0.000 .029577 .047770 .026702 .043319 .0098536 .0174177 .0212567 .0356291
P  0.000 .000230 .000409 .000104 .000186 .0000764 .0001489 .0001650 .0003050
S  0.000 .013211 .024358 .012780 .023667 .0044013 .0088810 .0094951 .0181672
Cl 0.000 .000155 .000316 .000070 .000143 .0000518 .0001155 .0001114 .0002358
Ar +.050 .001552 .003566 .000702 .001620 .0021275 .0053490 .0017949 .0042792
K  0.000 .000105 .000236 .000047 .000106 .0000346 .0000852 .0000753 .0001757
Ca 0.000 .001783 .004110 .002551 .005906 .0005935 .0014971 .0012813 .0030648
Ti 0.000 .000071 .000195 .000133 .000368 .0000230 .0000694 .0000506 .0001447
Cr 0.000 .000398 .001191 .000180 .000541 .0001325 .0004336 .0002862 .0008880
Mn 0.000 .000270 .000852 .000122 .000387 .0000898 .0003104 .0001938 .0006354
Fe 0.000 .025839 .082984 .011691 .037714 .0086083 .0302574 .0185702 .0618933
Ni 0.000 .001482 .005003 .000700 .002373 .0004940 .0018252 .0010653 .0037319
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .109806 .220345 .092864 .168505 .0381909 .0843862 .0795953 .1659613
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04hiCNOx15Ne.775992amet)

 Z/X = 0.01965326    AGS04hiCNOx2Ne_OFe.5_W95           AGS04hiCNOx2Ne.2amet
        AGS04hiCNOx2Ne        ||        AGS04hiCNOx2Ne.5amet     ||
   ==================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.050 .243112 .167004 .110446 .076243 .3098835 .2325011 .2712458 .1932028
N  +.060 .061067 .048920 .027742 .022333 .0821687 .0718935 .0699583 .0581094
O  +.050 .452697 .414238 .650349 .598029 .3872014 .3869797 .4250999 .4033349
Ne +.351 .137081 .158205 .121416 .140815 .1882627 .2373075 .1586465 .1898460
Na 0.000 .001644 .002161 .000747 .000987 .0004782 .0006868 .0011525 .0015713
Mg 0.000 .029909 .041576 .034130 .047677 .0087044 .0132154 .0209747 .0302317
Al 0.000 .002376 .003666 .001079 .001673 .0006916 .0011656 .0016661 .0026658
Si 0.000 .028563 .045881 .025891 .041793 .0083129 .0145842 .0200310 .0333623
P  0.000 .000222 .000393 .000101 .000180 .0000646 .0001249 .0001556 .0002858
S  0.000 .012759 .023394 .012393 .022836 .0037125 .0074350 .0089470 .0170104
Cl 0.000 .000150 .000304 .000068 .000139 .0000440 .0000975 .0001053 .0002214
Ar +.050 .001499 .003425 .000681 .001564 .0020588 .0051375 .0017349 .0041100
K  0.000 .000101 .000227 .000046 .000103 .0000294 .0000717 .0000711 .0001649
Ca 0.000 .001722 .003948 .002473 .005697 .0005009 .0012541 .0012077 .0028704
Ti 0.000 .000069 .000188 .000130 .000358 .0000197 .0000589 .0000480 .0001363
Cr 0.000 .000385 .001144 .000175 .000523 .0001118 .0003631 .0002697 .0008316
Mn 0.000 .000260 .000819 .000118 .000373 .0000760 .0002609 .0001829 .0005958
Fe 0.000 .024953 .079702 .011336 .036386 .0072621 .0253344 .0174992 .0579549
Ni 0.000 .001431 .004805 .000679 .002291 .0004168 .0015282 .0010038 .0034943
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .106043 .211633 .090047 .162580 .0324837 .0713182 .0750495 .1555069
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04hiCNOx2Ne.727860amet)

 Z/X = 0.02043057    AGS04hiCNOx25Ne_OFe.5_W95         AGS04hiCNOx25Ne.2amet
        AGS04hiCNOx25Ne       ||       AGS04hiCNOx25Ne.5amet     ||
   ==================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.050 .235057 .160650 .107192 .073650 .2992678 .2229701 .2620815 .1855780
N  +.060 .059044 .047059 .026925 .021574 .0795330 .0691020 .0676675 .0558762
O  +.050 .437697 .398479 .631190 .577693 .3661031 .3633412 .4075643 .3844241
Ne +.448 .165674 .190232 .147299 .170033 .2279647 .2853480 .1918908 .2282784
Na 0.000 .001589 .002079 .000725 .000953 .0003954 .0005638 .0010867 .0014729
Mg 0.000 .028918 .039994 .033124 .046055 .0071915 .0108424 .0197738 .0283333
Al 0.000 .002297 .003527 .001047 .001616 .0005719 .0009571 .0015710 .0024990
Si 0.000 .027617 .044135 .025128 .040371 .0068680 .0119652 .0188840 .0312671
P  0.000 .000214 .000378 .000098 .000174 .0000533 .0001024 .0001467 .0002678
S  0.000 .012336 .022504 .012028 .022059 .0030673 .0061000 .0084349 .0159424
Cl 0.000 .000145 .000292 .000066 .000134 .0000361 .0000795 .0000990 .0002070
Ar +.050 .001449 .003295 .000661 .001511 .0019945 .0049425 .0016789 .0039540
K  0.000 .000098 .000218 .000045 .000101 .0000240 .0000582 .0000669 .0001541
Ca 0.000 .001665 .003798 .002401 .005505 .0004139 .0010291 .0011386 .0026904
Ti 0.000 .000066 .000181 .000125 .000343 .0000163 .0000484 .0000453 .0001279
Cr 0.000 .000372 .001100 .000170 .000506 .0000921 .0002971 .0002541 .0007788
Mn 0.000 .000252 .000787 .000115 .000361 .0000625 .0002129 .0001721 .0005574
Fe 0.000 .024126 .076670 .011002 .035148 .0060003 .0207864 .0164976 .0543165
Ni 0.000 .001384 .004622 .000659 .002213 .0003443 .0012537 .0009463 .0032747
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .102528 .203580 .087394 .157050 .0271314 .0592387 .0707959 .1458433
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04hiCNOx25Ne.682227amet)

 Z/X = 0.02120788    AGS04hiCNOx3Ne_OFe.5_W95           AGS04hiCNOx3Ne.2amet
        AGS04hiCNOx3Ne        ||        AGS04hiCNOx3Ne.5amet     ||
   ==================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.050 .227518 .154762 .104125 .071229 .2892973 .2141381 .2534926 .1785124
N  +.060 .057150 .045334 .026155 .020865 .0770566 .0665145 .0655198 .0538062
```

```
O  +.050 .423658 .383873 .613128 .558696 .3462824 .3414322 .3911257 .3668969
Ne +.527 .192433 .219912 .171701 .197330 .2652590 .3298680 .2230522 .2638944
Na 0.000 .001538 .002003 .000704 .000922 .0003175 .0004498 .0010251 .0013817
Mg 0.000 .027991 .038528 .032177 .044541 .0057706 .0086434 .0186483 .0265741
Al 0.000 .002223 .003397 .001017 .001563 .0004583 .0007621 .0014811 .0023430
Si 0.000 .026731 .042518 .024409 .039044 .0055117 .0095397 .0178097 .0293267
P  0.000 .000207 .000364 .000095 .000168 .0000426 .0000814 .0001382 .0002510
S  0.000 .011940 .021679 .011683 .021332 .0024611 .0048625 .0079547 .0149524
Cl 0.000 .000140 .000282 .000064 .000129 .0000295 .0000645 .0000938 .0001950
Ar +.050 .001403 .003174 .000642 .001461 .0019339 .0047610 .0016262 .0038088
K  0.000 .000095 .000210 .000043 .000096 .0000192 .0000462 .0000630 .0001445
Ca 0.000 .001612 .003658 .002332 .005323 .0003316 .0008191 .0010734 .0025224
Ti 0.000 .000064 .000174 .000121 .000330 .0000128 .0000379 .0000426 .0001195
Cr 0.000 .000360 .001060 .000165 .000489 .0000740 .0002371 .0002397 .0007308
Mn 0.000 .000244 .000759 .000112 .000350 .0000505 .0001709 .0001626 .0005238
Fe 0.000 .023353 .073860 .010687 .033992 .0048149 .0165714 .0155587 .0509445
Ni 0.000 .001340 .004453 .000640 .002140 .0002765 .0010002 .0008926 .0030719
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .099241 .196119 .084891 .151880 .0221047 .0480472 .0668097 .1368901
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04hiCNOx3Ne.639022amet)

 Z/X = 0.02070490    AGS04vhCNOx15Ne_OFe.5_W95         AGS04vhCNOx15Ne.2amet
        AGS04vhCNOx15Ne       ||       AGS04vhCNOx15Ne.5amet     ||
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.100 .254360 .177864 .114980 .080443 .3236246 .2487911 .2836549 .2062348
N  +.120 .065381 .053315 .029554 .024112 .0875472 .0784860 .0747562 .0633834
O  +.100 .473639 .441177 .677043 .630963 .4173540 .4273882 .4498350 .4356617
Ne +.276 .107564 .126365 .094797 .111424 .1467589 .1895475 .1241412 .1516380
Na 0.000 .001533 .002051 .000693 .000928 .0003546 .0005218 .0010342 .0014393
Mg 0.000 .027890 .039464 .031667 .044832 .0064587 .0100474 .0188256 .0276973
Al 0.000 .002215 .003480 .001001 .001573 .0005134 .0008866 .0014955 .0024426
Si 0.000 .026635 .043550 .024023 .039300 .0061680 .0110877 .0179784 .0305651
P  0.000 .000207 .000373 .000094 .000170 .0000479 .0000949 .0001396 .0002618
S  0.000 .011897 .022206 .011498 .021472 .0027549 .0056530 .0080305 .0155848
Cl 0.000 .000140 .000289 .000063 .000130 .0000331 .0000750 .0000948 .0002034
Ar +.100 .001568 .003648 .000709 .001650 .0021401 .0054720 .0018103 .0043776
K  0.000 .000095 .000215 .000043 .000098 .0000215 .0000537 .0000636 .0001505
Ca 0.000 .001606 .003747 .002295 .005358 .0003713 .0009526 .0010837 .0026292
Ti 0.000 .000064 .000178 .000120 .000335 .0000143 .0000439 .0000429 .0001243
Cr 0.000 .000359 .001086 .000162 .000491 .0000830 .0002761 .0002421 .0007620
Mn 0.000 .000243 .000777 .000110 .000352 .0000563 .0001979 .0001640 .0005454
Fe 0.000 .023269 .075654 .010518 .034215 .0053889 .0192624 .0157065 .0530973
Ni 0.000 .001335 .004561 .000630 .002154 .0003093 .0011622 .0009010 .0032015
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .099056 .201279 .083626 .153058 .0247153 .0557872 .0676127 .1430821
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04vhCNOx15Ne.663458amet)

 Z/X = 0.02157703    AGS04vhCNOx2Ne_OFe.5_W95           AGS04vhCNOx2Ne.2amet
        AGS04vhCNOx2Ne        ||        AGS04vhCNOx2Ne.5amet     ||
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.100 .245556 .170675 .111458 .077562 .3121591 .2380076 .2736893 .1976080
N  +.120 .063118 .051160 .028649 .023249 .0846360 .0752535 .0722074 .0607974
O  +.100 .457245 .423344 .656303 .608366 .3944695 .4006387 .4307287 .4142621
Ne +.401 .138454 .161677 .122524 .143245 .1893233 .2425155 .1599419 .1940124
Na 0.000 .001480 .001968 .000672 .000895 .0002722 .0003973 .0009694 .0013397
Mg 0.000 .026925 .037869 .030698 .043228 .0049614 .0076549 .0176472 .0257833
Al 0.000 .002139 .003339 .000971 .001518 .0003942 .0006751 .0014017 .0022734
```

```
Si 0.000 .025713 .041790 .023287 .037893 .0047383 .0084477 .0168531 .0284531
P  0.000 .000200 .000358 .000091 .000163 .0000368 .0000724 .0001309 .0002438
S  0.000 .011486 .021309 .011147 .020705 .0021165 .0043075 .0075282 .0145084
Cl 0.000 .000135 .000277 .000061 .000125 .0000253 .0000570 .0000887 .0001890
Ar +.100 .001514 .003500 .000687 .001590 .0020703 .0052500 .0017490 .0042000
K  0.000 .000091 .000206 .000041 .000093 .0000162 .0000402 .0000594 .0001397
Ca 0.000 .001550 .003596 .002224 .005164 .0002854 .0007261 .0010161 .0024480
Ti 0.000 .000062 .000171 .000117 .000325 .0000110 .0000334 .0000403 .0001159
Cr 0.000 .000346 .001042 .000157 .000473 .0000637 .0002101 .0002269 .0007092
Mn 0.000 .000235 .000746 .000107 .000341 .0000434 .0001514 .0001539 .0005082
Fe 0.000 .022463 .072596 .010196 .032990 .0041396 .0146754 .0147232 .0494277
Ni 0.000 .001288 .004377 .000610 .002075 .0002378 .0008862 .0008447 .0029807
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .095627 .193144 .081066 .147578 .0194121 .0435847 .0634327 .1333201
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04vhCNOx2Ne.621160amet)

 Z/X = 0.02244915    AGS04vhCNOx25Ne_OFe.5_W95         AGS04vhCNOx25Ne.2amet
        AGS04vhCNOx25Ne       ||       AGS04vhCNOx25Ne.5amet     ||
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.100 .237341 .164045 .108145 .074880 .3014154 .2280626 .2643742 .1896520
N  +.120 .061006 .049172 .027797 .022445 .0819071 .0722715 .0698241 .0584118
O  +.100 .441949 .406897 .636799 .587335 .3730252 .3759682 .4128692 .3945257
Ne +.498 .167278 .194245 .148603 .172865 .2292092 .2913675 .1934070 .2330940
Na 0.000 .001430 .001892 .000652 .000864 .0001956 .0002833 .0009093 .0012485
Mg 0.000 .026024 .036398 .029785 .041732 .0035585 .0054484 .0165456 .0240181
Al 0.000 .002067 .003210 .000942 .001465 .0002833 .0004816 .0013147 .0021186
Si 0.000 .024853 .040167 .022595 .036583 .0033987 .0060132 .0158014 .0265055
P  0.000 .000193 .000344 .000088 .000157 .0000263 .0000514 .0001227 .0002270
S  0.000 .011101 .020481 .010815 .019988 .0015178 .0030655 .0070581 .0135148
Cl 0.000 .000130 .000266 .000059 .000121 .0000181 .0000405 .0000830 .0001758
Ar +.100 .001463 .003364 .000667 .001536 .0020051 .0050460 .0016919 .0040368
K  0.000 .000088 .000198 .000040 .000090 .0000114 .0000282 .0000557 .0001301
Ca 0.000 .001498 .003456 .002158 .004986 .0002044 .0005161 .0009525 .0022800
Ti 0.000 .000060 .000164 .000113 .000312 .0000076 .0000229 .0000376 .0001075
Cr 0.000 .000335 .001001 .000153 .000459 .0000454 .0001486 .0002125 .0006600
Mn 0.000 .000227 .000717 .000104 .000329 .0000312 .0001079 .0001443 .0004734
Fe 0.000 .021712 .069776 .009893 .031850 .0029690 .0104454 .0138042 .0460437
Ni 0.000 .001245 .004207 .000592 .002003 .0001707 .0006312 .0007920 .0027767
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
h>       .092426 .185641 .078656 .142475 .0144431 .0323302 .0595255 .1243165
    ==================== =============== ================= =================
 (NOTE: some abundances would go negative beyond AGS04vhCNOx25Ne.580960amet)

 Z/X = 0.02332128    AGS04vhCNOx3Ne_OFe.5_W95           AGS04vhCNOx3Ne.2amet
        AGS04vhCNOx3Ne        ||        AGS04vhCNOx3Ne.5amet     ||
   ===================== =============== ================= =================
i d:Ab_i  Ni/Nz   Xi/Z    Ni/Nz   Xi/Z     Ni/Nz    Xi/Z     Ni/Nz    Xi/Z
-- ----- ------- ------- ------- ------- -------- -------- -------- --------
C  +.100 .229658 .157910 .105024 .072378 .2913253 .2188601 .2556460 .1822900
N  +.120 .059031 .047333 .026995 .021695 .0793452 .0695130 .0675919 .0562050
O  +.100 .427642 .391682 .618420 .567711 .3528913 .3531457 .3961398 .3762677
Ne +.577 .194235 .224377 .173177 .200507 .2666617 .3365655 .2247580 .2692524
Na 0.000 .001384 .001821 .000633 .000835 .0001230 .0001768 .0008523 .0011633
Mg 0.000 .025181 .035037 .028925 .040337 .0022411 .0034069 .0155137 .0223849
Al 0.000 .002000 .003090 .000915 .001417 .0001787 .0003016 .0012327 .0019746
Si 0.000 .024048 .038665 .021942 .035359 .0021405 .0037602 .0148158 .0247031
P  0.000 .000187 .000331 .000086 .000153 .0000165 .0000319 .0001150 .0002114
S  0.000 .010742 .019715 .010503 .019320 .0009557 .0019165 .0066178 .0125956
Cl 0.000 .000126 .000256 .000057 .000116 .0000115 .0000255 .0000778 .0001638
Ar +.100 .001416 .003238 .000648 .001485 .0019439 .0048570 .0016384 .0038856
```

```
 K  0.000 .000085 .000191 .000039 .000087 .0000072 .0000177 .0000524 .0001217
 Ca 0.000 .001450 .003327 .002096 .004820 .0001287 .0003226 .0008932 .0021252
 Ti 0.000 .000058 .000158 .000110 .000302 .0000046 .0000139 .0000353 .0001003
 Cr 0.000 .000324 .000964 .000148 .000442 .0000286 .0000931 .0001994 .0006156
 Mn 0.000 .000219 .000690 .000100 .000315 .0000196 .0000674 .0001352 .0004410
 Fe 0.000 .021009 .067166 .009607 .030784 .0018695 .0065304 .0129429 .0429117
 Ni 0.000 .001205 .004049 .000575 .001937 .0001074 .0003942 .0007424 .0025871
 -- ----- ------- ------- ------- ------- -------- -------- -------- --------
 h>       .089434 .178698 .076384 .137709 .0097765 .0219157 .0558643 .1159849
     =============== =============== ================= =================
  (NOTE: some abundances would go negative beyond AGS04vhCNOx3Ne.548057amet)

-------------------------------------------------------------------------------
  OPAL ATOMIC MASSES:  We list the OPAL atomic masses (atomic weights) used
  to convert number fractions (Ni) to mass fractions (X, Y, Z, Xi).

  Element  Mass (a.u.)    Element  Mass (a.u.)    Element  Mass (a.u.)
  -------  -----------    -------  -----------    -------  -----------
   1  H       1.00790     12  Mg     24.30500     19  K      39.09830
   2  He      4.00260     13  Al     26.98154     20  Ca     40.08000
   6  C      12.01100     14  Si     28.08550     22  Ti     47.90000
   7  N      14.00670     15  P      30.97376     24  Cr     51.99600
   8  O      15.99940     16  S      32.06000     25  Mn     54.93800
  10  Ne     20.17900     17  Cl     35.45300     26  Fe     55.84700
  11  Na     22.98977     18  Ar     39.94800     28  Ni     58.70000
-------------------------------------------------------------------------------


         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         Details of the accuracy of the X-, Z-, and CO-interpolation
         ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

  Z-interpolation errors if 'GN93hz' file not used:
  ==================================================

  For the non-CO-rich mixes (C=O=0.0), one can check the accuracy of the
  Z-interpolation among the 'Gz???.x??' files when opacities are read in, by
  looking at the 'GN93hz' opacities.  Where the Z values are in both files,
  the opacities are identical.  When only 'GN93hz' (or only the 'Gz???.x??'
  files) contains the Z value, errors in interpolation among 'Gz???.x??' files
  are shown below; the largest, the mean, and the rms error in logKappa are
  given (at T6 of 0.01 or higher, and all log R values).  Note that the
  'Gz???.x??' files contain Z = 0.05, but 'GN93hz' does not; for this case,
  interpolation in 'GN93hz' is tested, rather than interpolation among
  'Gz???.x??' files.  Note that the Z-interpolation errors tend to be quite
  small, with an rms error of less than 4% even in the worst case; applying
  the 'GN93hz' opacity-shifts (as is the default) when interpolating in Z
  should significantly reduce these errors (note: T6 < 0.01 omitted):

  Z-interpolation errors if 'GN93hz' file not used (if khighz=0 in READZEXCO):
  ----------------------------------------------------------------------------

       dLogKappa(X=0.00) dLogKappa(X=0.10) dLogKappa(X=0.35) dLogKappa(X=0.70)
       ================= ================= ================= =================
   Z      max  mean   rms   max  mean   rms   max  mean   rms   max  mean   rms
 -----  ---- ----- -----  ---- ----- -----  ---- ----- -----  ---- ----- -----
 .0001  .128 .0078 .0169  .042 .0047 .0084  .040 .0041 .0077  .039 .0040 .0078
 .0003  .112 .0072 .0151  .045 .0046 .0084  .043 .0041 .0079  .046 .0040 .0081
 .002  -.036-.0018 .0038 -.012-.0012 .0023 -.012-.0011 .0022 -.013-.0010 .0025
 .04   -.003 .0000 .0003 -.001 .0000 .0002  .001 .0000 .0002  .001 .0000 .0002
 .05 * -.003 .0000 .0003  .001 .0000 .0002  .001 .0000 .0002  .001 .0000 .0002
 .06   -.004 .0000 .0004 -.001 .0000 .0003  .001 .0000 .0003 -.001 .0000 .0003
 .08   -.003 .0000 .0004 -.001 .0000 .0003  .001 .0000 .0003 -.001 .0000 .0003
```

It is clear from the following table that X-interpolation errors in the file 'GN93hz' (to get X=0.03) would be much larger than any Z-interpolation error in the files Gz???.x03 ; thus any opacity shifts for X=0.03 are interpolated from the X=0, X=0.1, and X=0.35 opacity shifts (unless only a single mix is being read in, which is NOT the default case). The size of the error in these X-interpolated opacity shifts is presumably somewhat smaller than the opacity shifts themselves, which in turn are smaller than the errors shown below that would result if the 'GN93hz' opacities were interpolated in X to get the X=0.03 opacities.

X-interpolation errors at X=0.03 if ONLY the 'GN93hz' file were used:
==

X interpolation errors, for X = 0.03, interpolating in X = 0.0, 0.1, 0.2 in the file 'GN93hz' (note that a value of Xdel = 0.005 was used for this interpolation, and all T6 < 0.01 opacities were omitted):

X-interpolation errors that would occur if 'Gz???.x??' files were not used:

Z:	0.	0.001	0.004	0.01	0.02	0.03	0.1	(at X=0.03)
max	-.3514	-.2971	-.2454	-.1913	-.1396	-.1062	-.0614	(dLogKappa)
mean	-.0159	-.0149	-.0141	-.0132	-.0124	-.0118	-.0103	
rms	.0399	.0350	.0305	.0259	.0219	.0196	.0158	

X-interpolation/extrapolation errors if 'GN93hz' file not used:
==

The file 'GN93hz' contains (non-CO-rich) opacities at X-values not available from the 'Gz???.x??' files, namely, X = 0.2, 0.5, 0.8, 0.9, 0.95, and 1-Z. If one sets khighz = 0 in the call to READZEXCO that reads the opacities, then the 'GN93hz' file is not read in and X-interpolation is less accurate [or alternatively, if one turns off "accurate-X" by calling SET_XHI(0)]. For X < 0.75 or so, the errors are comparable to or smaller than the errors from the original OPAL opacity computation; but for extrapolation to larger X-values, the error grows very rapidly, and can become as large as an order of magnitude as X approaches 1-Z:

X-interpolation/extrapolation errors if 'GN93hz' file not used (khighz=0):

***Interpolation (dLogKappa):

Z:	0.	0.0001	0.001	0.004	0.01	0.02	0.03	0.05	0.08	0.1
X=0.2:										
max	-.0146	.0375	-.0153	-.0147	-.0139	-.0137	-.0142	-.0129	-.0124	-.0126
mean	-.0023	-.0017	-.0021	-.0018	-.0016	-.0014	-.0013	-.0012	-.0010	-.0010
rms	.0044	.0051	.0040	.0036	.0033	.0031	.0029	.0027	.0025	.0024
X=0.5:										
max	.0291	.0291	.0284	.0269	.0260	.0249	.0244	.0243	.0234	.0234
mean	.0028	.0027	.0023	.0019	.0016	.0013	.0011	.0010	.0008	.0008
rms	.0076	.0073	.0066	.0059	.0053	.0048	.0045	.0042	.0039	.0038

***Extrapolation (dLogKappa):

Z:	0.	0.0001	0.001	0.004	0.01	0.02	0.03	0.05	0.08	0.1
X=0.8:										
max	-.0732	-.0721	-.0706	-.0680	-.0637	-.0620	-.0597	-.0578	-.0557	-.0565
mean	-.0071	-.0068	-.0059	-.0049	-.0041	-.0035	-.0031	-.0027	-.0024	-.0023
rms	.0178	.0172	.0153	.0134	.0119	.0107	.0100	.0092	.0086	.0084
X=0.9:										

```
   max  −.2415 −.2405 −.2328 −.2220 −.2063 −.1980 −.1914 −.1854 −.1828 −.1860
  mean  −.0226 −.0216 −.0186 −.0154 −.0129 −.0110 −.0099 −.0088 −.0082 −.0085
   rms   .0565  .0544  .0480  .0416  .0366  .0328  .0306  .0284  .0273  .0281
X=0.95: −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−−
   max  −.4256 −.4189 −.4025 −.3805 −.3555 −.3321 −.3264 −.3213
  mean  −.0377 −.0360 −.0307 −.0253 −.0212 −.0181 −.0165 −.0154
   rms   .0950  .0910  .0797  .0684  .0600  .0537  .0505  .0486
X=1−Z:  −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−−
   max  −.9901 −.9632 −.8694 −.7293 −.6045 −.4773 −.4059 −.3213 −.2219 −.1860
  mean  −.0818 −.0758 −.0609 −.0468 −.0370 −.0306 −.0279 −.0154 −.0175 −.0085
   rms   .2079  .1941  .1592  .1249  .0982  .0769  .0645  .0486  .0356  .0281
        −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−− −−−−−−
```

CO−interpolation errors:

Note that there are six cases (at three different metallicities) where mixes on the line C+O = 1−X−Z, with identical compositions, are interpolated in two different ways when being read into two different positions in the matrix CO; since these cases all have X > 0, they do not correspond to mixes that are likely to be encountered by investigators, but they do give another estimate of interpolation errors for some CO−rich mixes. (Note that there are several other cases where mixes with identical compositions appear at two different places in the matrix CO, but these occur at tabulated Z−values and thus have identical opacity values.) Differences for the CO−rich cases with identical compositions (note that T6 < 0.01 were omitted):

Differences for CO−rich mixes interpolated in two different ways:

```
 case:   Z=0.04 X=0.35   Z=0.07 X=0.7    Z=0.09 X=0.7
        ==============  ==============  ==============
C & O:  .01 .6  .6 .01  .03 .2  .2 .03  .01 .2  .2 .01
        −−−−−−  −−−−−−  −−−−−−  −−−−−−  −−−−−−  −−−−−−
   max  −.0333  −.0026  −.0491  −.0110  −.0422  −.0084  (dLogKappa)
  mean  −.0029  −.0002  −.0046  −.0014  −.0041  −.0011
   rms   .0061   .0008   .0091   .0029   .0081   .0023
```

These errors are still quite small, with an rms of 1% or less, smaller than the estimated error in the opacity computations quoted above or than the largest of the Z−interpolation errors at C=O=0.0 (though larger than the errors for the C=O=0.0 mix for the same metallicity Z). The maximum error is less than 12% (note that the maximum errors tend to lie at fairly low temperatures, where CO−rich opacities are less likely to be needed).

The above tables of errors were obtained by considering Z−interpolation as the opacities were read in. One may also compare opacities where the only Z−interpolation was performed on input (i.e., Nzin = 1 in READZEXCO) with opacities interpolated in Z by the call to OPAC or OPAL (i.e., which had had Z > 4 in READZEXCO). These are interpolated among somewhat different grid points, and thus give some idea of the Z−interpolation and CO−interpolation errors. NOTE THAT USE OF COINTERP WILL LEAD TO SIGNIFICANTLY LARGER ERRORS, as discussed further below. For several Z−values, the number of points compared and the maximum and rms differences in log10(Kappa) are given below both at and between (X,T6,R,C,O) gridpoints, for low (4<logT<6) and high (logT>6) temperatures (T6 < 0.01 are omitted; table Z−values are also omitted, since opacity differences are zero there, as expected). Note that rms differences are always small, less than 0.3%, but this may be misleading as many comparison points will interpolate between the same gridpoints, just in a different order, and thus will have identical interpolated log10(Kappa) values. For C = O = 0.0, the maximum differences are small, less than 1%; however, the CO−interpolation can induce somewhat larger errors in opacities

```
 of CO-rich mixes at low metallicity: for logT > 6 at X = 0.0 (where CO-rich
 opacities are likely to be needed), the maximum differences do not exceed
 1% for Z > 0.001, but they can be as high as 8% for 0.0001 < Z < 0.001, and
 can reach 11% for Z < 0.0001.

 "max" gives some indications of combined Z- and CO-interpolation errors:
 ------------------------------------------------------------------------


    dLogKappa for:                        <-------- max{C,O} > 0.0 --------->
      <------ C = O = 0.0 , all X ------> <----On-Grid----> <----Off-Grid--->
      <----On-Grid----> <----Off-Grid--->   all X   X = 0.0   all X   X = 0.0
  Z   4<logT<6   logT>6 4<logT<6   logT>6   logT>4   logT>6   logT>4   logT>6
===== -------- -------- -------- -------- -------- -------- -------- --------
.00001:N= 3895     2075    27695    15375   273426    21580  7185558   315535
  max -.000038  .000067 -.000069  .000121 -.066377  .009322 -.093110  .012053
  rms  .000003  .000004  .000004  .000004  .000178  .000103  .000174  .000074
----- -------- -------- -------- -------- -------- -------- -------- --------
.00005:N= 3895     2075    27695    15375   279396    21995  7228628   314790
  max -.000080  .000171 -.000156  .000310  .046540  .039973 -.118741  .046853
  rms  .000008  .000009  .000010  .000010  .000401  .000363  .000352  .000230
.0002:-------- -------- -------- -------- -------- -------- -------- --------
  max -.000214  .000447 -.000404  .000828 -.041788  .028401 -.058897 -.035296
  rms  .000010  .000015  .000012  .000016  .000365  .000267  .000317  .000197
.0005:-------- -------- -------- -------- -------- -------- -------- --------
  max -.000316 -.000304 -.000512  .000854 -.044971  .017734 -.061407  .020958
  rms  .000025  .000025  .000032  .000030  .000357  .000155  .000271  .000121
.0015:-------- -------- -------- -------- -------- -------- -------- --------
  max  .001481 -.000494  .002325  .001390 -.019924  .001625 -.025380  .001904
  rms  .000086  .000033  .000111  .000042  .000202  .000062  .000165  .000059
.0030:-------- -------- -------- -------- -------- -------- -------- --------
  max -.001855 -.000593 -.002934 -.001971 -.006332 -.001797  .006585 -.002046
  rms  .000107  .000032  .000138  .000049  .000181  .000076  .000158  .000072
----- -------- -------- -------- -------- -------- -------- -------- --------
.0070: N= 3895     2075    27695    15375   279396    21995  7218820   314790
  max  .000882  .000475  .001406 -.001663 -.005493  .001000 -.008853  .001083
  rms  .000054  .000026  .000070  .000041  .000098  .000044  .000111  .000043
----- -------- -------- -------- -------- -------- -------- -------- --------
.0150: N= 3895     2075    27695    15375   279396    21995  7199204   314790
  max -.000052 -.000111 -.000136 -.000213 -.009724 -.000550  .038904 -.001056
  rms  .000002  .000010  .000003  .000009  .000087  .000019  .000482  .000019
----- -------- -------- -------- -------- -------- -------- -------- --------
.0250: N= 3895     2075    27695    15375   277008    21995  7137840   314790
  max  .000087 -.000088  .000233 -.000140 -.003307  .000414  .036001 -.000738
  rms  .000006  .000009  .000009  .000009  .000049  .000025  .000439  .000023
----- -------- -------- -------- -------- -------- -------- -------- --------
.0350: N= 3895     2075    27695    15375   277008    21995  7059376   314790
  max -.000253 -.000406  .002127  .000610 -.005123 -.000655 -.041995 -.000777
  rms  .000024  .000031  .000032  .000033  .000099  .000046  .000626  .000050
----- -------- -------- -------- -------- -------- -------- -------- --------
.0450: N= 3895     2075    27695    15375   274620    21995  7022532   314790
  max  .000980  .000782  .001798  .003339  .009452  .000765  .054504  .001237
  rms  .000061  .000076  .000069  .000090  .000149  .000065  .000921  .000080
----- -------- -------- -------- -------- -------- -------- -------- --------
.0550: N= 3895     2075    27695    15375   268650    21995  6940230   311300
  max  .000457 -.000596  .001611 -.001637  .004770 -.000654  .006170  .001523
  rms  .000043  .000055  .000047  .000060  .000081  .000045  .000108  .000041
----- -------- -------- -------- -------- -------- -------- -------- --------
.0700: N= 3895     2075    27695    15375   266262    21995  6844538   311300
  max -.000404 -.000476  .001360 -.001767  .007161 -.000599  .042980 -.001666
  rms  .000027  .000038  .000032  .000046  .000175  .000036  .000708  .000097
----- -------- -------- -------- -------- -------- -------- -------- --------
.0900: N= 3895     2075    27695    15375   263874    21995  6665478   311300
  max -.000236  .000500 -.001106  .002409  .008298  .000866 -.045470  .004078
  rms  .000022  .000042  .000034  .000066  .000221  .000029  .000966  .000140
----- -------- -------- -------- -------- -------- -------- -------- --------
```

Z-interpolation errors when one uses linear Z-interpolation (Nzin = 2):
===

If, for example, diffusion leads to relatively small Z-variations in a star (say, of order 10%), then one might wish to use linear interpolation in logZ by setting nZin = 2 in READZEXCO. (Note that opacity interpolation with nZin = 2 will usually be significantly faster than nZin = 3 , which will in general be faster than nZin = 4 ; however, all values of nZin > 4 take the same amount of time as the nZin = 4 case, except for the slight added start-up time to read in the extra opacity files). The size of the errors introduced by nZin = 2 linear interpolation are easily estimated, e.g., by comparing { nZin = 2, Zlo = .019, Zhi = .021 } opacities at Z = 0.02 with the Z=0.02 opacity tables themselves. Such errors for the nZin = 2 case are tabulated below for several values of Z, for two different cases with Z-ranges of +/- 10% and of +/- 20%, respectively (referred to as cases "1" and "2" in the table headings on the left). The mean errors are given as well as the maximum and rms errors, since it is not unreasonable to expect a systematic tendency from linear interpolation of a curve; however, they turn out to be negligible (always < 0.04%) for a Z-range of +/- 10%. The rms error in the Z-interpolation is likewise negligible (always < 0.2%) for a Z-range of +/- 10%; the maximum errors are less than 1% for C = O = 0.0, and also for Z > 0.001 at logT > 6 with X = 0.0, max{C,O} > 0.0, although for Z < 0.001 the CO-interpolation can result in errors up to 9% in a few places for CO-rich mixes. Even for a Z-range of +/- 20%, the mean errors are always < 0.14%; the C = O = 0.0 case has rms errors < 0.2% and maximum errors < 1.7%, while { Z > 0.001, logT > 6, X = 0.0, max{C,O} > 0.0 } has rms errors < 0.21% and maximum errors < 1.4% (up to 9% for Z < 0.001).

Errors if linear Z-interpolation is used, with Z-ranges +/-10% and +/-20%:

```
   Z= 0.0001   0.001    0.004    0.01     0.02     0.03     0.05     0.08
      ======== ======== ======== ======== ======== ======== ======== ========
Zlo1=  0.00009  0.0009   0.0036   0.009    0.018    0.027    0.045    0.072
Zhi1=  0.00011  0.0011   0.0044   0.011    0.022    0.033    0.055    0.088
----- -------- -------- -------- -------- -------- -------- -------- --------
Zlo2=  0.00008  0.0008   0.0032   0.008    0.016    0.024    0.040    0.064
Zhi2=  0.00012  0.0012   0.0048   0.012    0.024    0.036    0.060    0.096
      ======== ======== ======== ======== ======== ======== ======== ========
C=O=0.0, OnGrid, logT>4 dLogKappa:
 max1 -.000703  .001008  .001107  .000764  .000989  .000925 -.001984  .001149
mean1 -.000026  .000017  .000065  .000103  .000133  .000145  .000150  .000149
 rms1  .000056  .000064  .000109  .000146  .000183  .000199  .000241  .000214
----- -------- -------- -------- -------- -------- -------- -------- --------
 max2 -.002772  .001246 -.002859  .003076  .003924  .003731  .004402  .004059
mean2 -.000103  .000043  .000238  .000416  .000537  .000583  .000612  .000609
 rms2  .000219  .000145  .000375  .000587  .000733  .000793  .000857  .000853
      ======== ======== ======== ======== ======== ======== ======== ========
C=O=0.0, OffGrid, logT>4 dLogKappa:
 max1 -.000707 -.002721  .003203  .001427  .002620  .002294  .003412  .002120
mean1 -.000024  .000015  .000065  .000100  .000130  .000143  .000149  .000149
 rms1  .000051  .000076  .000119  .000141  .000178  .000197  .000245  .000213
----- -------- -------- -------- -------- -------- -------- -------- --------
 max2 -.002806 -.002473 -.005426  .005459  .005472  .003856  .006457  .007094
mean2 -.000095  .000041  .000232  .000405  .000524  .000573  .000606  .000605
 rms2  .000198  .000153  .000367  .000568  .000713  .000780  .000848  .000849
      ======== ======== ======== ======== ======== ======== ======== ========
max{C,O}>0.0, X=0.0, OnGrid, logT>6 dLogKappa:
 max1 -.032688  .000776  .000689 -.000687  .000569  .000581 -.001612  .000677
```

```
mean1 -.000003  .000007  .000030  .000065  .000098  .000115  .000148  .000147
 rms1  .000221  .000024  .000063  .000117  .000163  .000190  .000248  .000219
----- -------- -------- -------- -------- -------- -------- -------- --------
 max2 -.032967  .000873  .001402  .001871  .002107  .002235  .002763  .002591
mean2 -.000005  .000027  .000122  .000262  .000395  .000464  .000589  .000593
 rms2  .000223  .000071  .000247  .000468  .000651  .000749  .000905  .000875
      ======== ======== ======== ======== ======== ======== ======== ========
max{C,O}>0.0, X=0.0, OffGrid, logT>6 dLogKappa:
 max1 -.038349  .001053  .000879 -.001188 -.000875 -.001059 -.002342 -.002052
mean1 -.000002  .000007  .000029  .000064  .000097  .000114  .000148  .000148
 rms1  .000140  .000023  .000060  .000114  .000160  .000187  .000244  .000222
----- -------- -------- -------- -------- -------- -------- -------- --------
 max2 -.038750 -.001171  .001398  .002108  .002854 -.003313 -.004253 -.006058
mean2 -.000004  .000027  .000118  .000257  .000390  .000458  .000586  .000594
 rms2  .000156  .000071  .000239  .000458  .000641  .000740  .000898  .000881
      ======== ======== ======== ======== ======== ======== ======== ========
max{C,O}>0.0, all-X, OnGrid, logT>4 dLogKappa:
 max1 -.032688 -.025873 -.001446  .001413 -.003337  .003056 -.009148  .001397
mean1 -.000017  .000015  .000039  .000078  .000106  .000121  .000134  .000137
 rms1  .000111  .000135  .000075  .000118  .000159  .000179  .000261  .000198
----- -------- -------- -------- -------- -------- -------- -------- --------
 max2 -.036713 -.025350  .005615  .005444  .007725 -.008038 -.007972  .007035
mean2 -.000058  .000047  .000154  .000320  .000439  .000495  .000567  .000569
 rms2  .000237  .000228  .000250  .000463  .000615  .000697  .000813  .000804
      ======== ======== ======== ======== ======== ======== ======== ========
max{C,O}>0.0, all-X, OffGrid, logT>4 dLogKappa:
 max1 -.038349 -.034450 -.004328 -.004072 -.028706 -.028234 -.047400  .041999
mean1 -.000015  .000016  .000039  .000078  .000102  .000116  .000119  .000143
 rms1  .000073  .000101  .000075  .000116  .000377  .000375  .000776  .000729
----- -------- -------- -------- -------- -------- -------- -------- --------
 max2 -.050575 -.032346  .005371  .008759 -.027245  .052397 -.045954  .042481
mean2 -.000056  .000050  .000152  .000318  .000432  .000494  .000546  .000576
 rms2  .000193  .000174  .000245  .000461  .000692  .000952  .001109  .001302
      ======== ======== ======== ======== ======== ======== ======== ========
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Details of opacity shifts from initial smoothing when opacities are read in
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

The opacity tables to be interpolated among (i.e., the OPAL files) are known to have somewhat random numerical errors of a few percent. Consequently, adjusting the data prior to performing the interpolation is justified at this level. The code first reads the original (unsmoothed) tabular data; this data is then passed through a smoothing filter, using a set of routines developed by Mike Seaton (see M.J. Seaton, MNRAS 265, L25, 1993). It is the adjusted data that is actually used in carrying out the interpolations in OPAC or OPAL. The initial adjustment step helps improve the smoothness of the OPAC output, particularly at the smallest values of R. The medium to large R output is only slightly affected by this step. It takes only a few seconds to carry out the initial data smoothing step, but this initial smoothing can be skipped by calling the subroutine SET_SMOOTH (described further above) with a value of initsmooth = 0 .

In addition, a few opacities in the mixes adjacent to the C=O=0.0 mix (i.e., in the three mixes with C or O = 0.01, and C+O no more than 0.02) are smoothed in the C-O direction, if opacity changes between mixes with C,O = 0.0, 0.03, 0.1 are monotonic but the opacity at C,O = 0.01 does not fit the trend; the resulting adjustments are small, and only occur at a small minority of the (T6,R) points, but this smoothing can also be skipped, by calling SET_SMOOTH (described further above) with lowCOsmooth = 0 .

Maximum and rms differences between smoothed and unsmoothed opacity tables for selected metallicities Z, for non-CO-rich mixes ("CO=0") and CO-rich mixes ("CO>0") of each hydrogen abundance X, at intermediate temperatures ("4<logT<6") and high temperatures ("logT>6"); note: T6 < 0.01 was omitted:

```
Opacity shifts resulting from initial smoothing when they are read in:
----------------------------------------------------------------------


   dLogKappa for:
         X =    0.00         0.03         0.10         0.35         0.70
             ------------ ------------ ------------ ------------ ------------
               max   rms    max   rms    max   rms    max   rms    max   rms
Z=0.0: CO=0: ====== ===== ====== ===== ====== ===== ====== ===== ====== =====
    4<logT<6  .0288 .0062  .0492 .0082  .0534 .0072  .0278 .0044  .0168 .0026
      logT>6  .0172 .0023  .0175 .0022  .0160 .0021  .0125 .0016 -.0038 .0006
  CO>0:      ------ ----- ------ ----- ------ ----- ------ ----- ------ -----
    4<logT<6 -.0512 .0077  .0761 .0075  .0763 .0064  .0800 .0050  .0874 .0037
      logT>6  .0767 .0025  .0771 .0026  .0776 .0026  .0782 .0027  .0757 .0017
Z=.001:CO=0: ====== ===== ====== ===== ====== ===== ====== ===== ====== =====
    4<logT<6 -.0416 .0064  .0469 .0079  .0524 .0069  .0255 .0043  .0154 .0027
      logT>6 -.0149 .0022 -.0148 .0021 -.0145 .0020 -.0124 .0016  .0059 .0011
  CO>0:      ------ ----- ------ ----- ------ ----- ------ ----- ------ -----
    4<logT<6  .0792 .0072  .0689 .0072  .0528 .0061 -.0517 .0047  .0458 .0033
      logT>6 -.0324 .0023 -.0341 .0023 -.0362 .0024 -.0369 .0024 -.0153 .0015
Z=.02: CO=0: ====== ===== ====== ===== ====== ===== ====== ===== ====== =====
    4<logT<6 -.0570 .0069  .0483 .0076  .0517 .0066  .0200 .0043 -.0128 .0031
      logT>6 -.0108 .0021 -.0113 .0021 -.0110 .0021 -.0107 .0020 -.0083 .0017
  CO>0:      ------ ----- ------ ----- ------ ----- ------ ----- ------ -----
    4<logT<6  .0785 .0071  .0675 .0069  .0521 .0057 -.0493 .0044 -.0375 .0033
      logT>6 -.0303 .0023 -.0320 .0023 -.0345 .0024 -.0343 .0024 -.0140 .0016
Z=.10: CO=0: ====== ===== ====== ===== ====== ===== ====== ===== ====== =====
    4<logT<6 -.0397 .0073  .0564 .0078  .0496 .0066  .0182 .0047 -.0137 .0038
      logT>6 -.0093 .0021 -.0096 .0021 -.0097 .0021 -.0096 .0021 -.0087 .0018
  CO>0:      ------ ----- ------ ----- ------ ----- ------ ----- ------ -----
    4<logT<6  .0765 .0074  .0664 .0070  .0497 .0058 -.0404 .0046 -.0259 .0038
      logT>6 -.0246 .0023 -.0262 .0023 -.0273 .0024 -.0259 .0023 -.0095 .0017
```

For T6 > 0.01, the rms effect of the smoothing is always less than 2%, i.e., comparable to the Z-interpolation errors found for the CO-rich mixes above, and smaller than the estimated opacity computation errors.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Details of differences in CO-rich opacities from COINTSMO vs. COINTERP
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Use of the smoother CO-interpolation routine COINTSMO (rather than the old routine COINTERP) yields opacities that differ at only a few grid-points (those which COINTERP ignores when interpolating opacities), but that differ over a significant area of the CO-plane between grid-points. Opacities were compared at points chosen randomly in log T, log RHO, C, and O (always with C+O > 0, and with some excess probability of having either C=0, O=0, or C+O=1-X-Z). Opacity differences are tabulated below for selected metallicities Z, for X = 0 and for two ranges of non-zero X, at intermediate temperatures ("4<logT<6") and high temperatures ("logT>6"); note that very low temperatures (T6 < 0.01) were omitted:

```
CO-interpolation differences: from using subroutines COINTSMO vs. COINTERP:
---------------------------------------------------------------------------


dLogKappa for:       X = 0.0           0.0 < X < 0.35        0.35 < X < 0.8
               -------------------  -------------------  -------------------
                   N     max   rms      N     max   rms      N     max   rms
Z=0.0:         ====== ====== =====  ====== ====== =====  ====== ====== =====
     4<logT<6  209131 -.0495 .0021   91697 -.0540 .0042  117239 -.0939 .0061
```

logT>6	252334	−.0198	.0009	109760	−.0575	.0045	142313	.1272	.0062
Z=0.001:	======	======	=====	======	======	=====	======	======	=====
4<logT<6	208824	−.0376	.0019	91547	−.0525	.0040	117033	.0872	.0056
logT>6	251965	−.0195	.0008	109603	−.0568	.0045	142087	.1259	.0061
Z=0.02:	======	======	=====	======	======	=====	======	======	=====
4<logT<6	208225	−.0275	.0014	91265	.0388	.0032	116669	.0975	.0042
logT>6	251235	−.0161	.0006	109266	.0538	.0039	141670	.1031	.0051
Z=0.1:	======	======	=====	======	======	=====	======	======	=====
4<logT<6	207954	−.0189	.0010	91162	.0318	.0024	116518	.0591	.0031
logT>6	250923	−.0104	.0003	109122	.0452	.0029	141482	.0812	.0039

The routine COINTERP may have opacity discontinuities of the same order as the opacity differences (up to 5%, for X=0 and logT > 6; larger elsewhere), at those points where it switches over from interpolation in one direction to interpolation in another direction, interpolating among a different set of gridpoints (this generally occurs somewhere in the region O = C +/− 0.2).

+++

Details of the individual OPAL opacity tables and program storage space

+++

Each of the individual tables in a file Gz???.x?? covers 70 temperatures in the range logT=3.75 [T6=0.0056341325] (referred to as temperature 1) to logT=8.7 [T6=501.187] (note that the logT step size is 0.05 below logT=6.0, 0.10 below logT=8.1, and 0.20 above that), and covers 19 values of log R in the range logR=−8.0 (referred to as 1) to logR=+1.0, at half−integer steps. (NOTE: earlier tables were explicitly in terms of T6. For convenience the present tables tabulate log Kappa vs logT. The interpolation however still uses logT6 for the temperature variable, not logT.) Values of Z from 0.0 to 0.1 are available (at up to 14 values of Z). Type 2 OPAL tables have 8 Z−values from Z = 0.0 to Z = 0.1 and 5 X−values from X = 0.0 to X = 0.7 (plus 8 each of excess−C and excess−O values from 0.0 to 1 − X − Z), while Type 1 OPAL tables have 13 Z−values from Z = 0.0 to Z = 0.1 and 10 X−values from X = 0.0 to X = 1 − Z (but with less−good X−interpolation at low X); combining these tables allows accurate X−interpolation at all X, and more accurate Z−interpolation among 14 Z−values.

The sizes of the matrices (holding the input opacities) are set by the constant values in parameter statements. The number NZ of available Z−storage values was mentioned above (in the discussion of the inputs to the subroutine READZEXCO); its value in the parameter statements can be changed to any value between 1 and 14 (provided that it is the same everywhere!) and the program recompiled. Smaller values of NZ yield smaller ranges where Z can be interpolated (or less accurate interpolation over a wide range), but also save storage space; NZ = 5 is a reasonable compromise. Low values (NZ = 2 or 3) yield less accurate interpolation, but reduce both the storage space and the typical amount of CPU−time per opacity interpolation, since fewer Z−grid values need to be computed in general. For NZ = 1, only a constant Z can be accomodated. Other than NZ, no size parameter of the OPAL matrices should be changed.

The molecular and conductive opacity matrix sizes are independent of NZ, taking up about 1 Mb; their sizes should never be changed. The Ferguson et al. (2005) molecular opacity tables have the same range in logR as the OPAL tables, but with 2.7 < logT < 4.5 (or possibly 2.8 < logT < 4.5 for some cases); they are available at 16 Z−values from Z = 0.0 to Z = 0.1, and at 10 X−values from X = 0.0 to X = 1 − Z. For low X, their X−interpolation is less accurate (but at low temperatures where molecular opacities are needed one seldom will encounter X values close to zero).

Roughly (1.1 Mb) + NZ * (1.63 Mb) of storage space is required in total.

****