Notes on Opacity-Interpolationroutine OPINT (v9, 20/08/97)

1. Introduction

This package interpolates opacities in ρ,T,X,Z (ρ = density, T = temperature, X = hydrogen mass fraction, Z = heavy elements mass fraction). In this package the latest 2 versions of the OPAL-tables [RI92, RI95] are used. One can choose between two algorithms. The first algorithm (minimum norm) uses a C^1 interpolant defined in piecewise fashion over triangles in the ρ,T - plane [MC89] [Nie83]. The global interpolant is constucted by a nine paramter version of a finite interpolation scheme introduced by Nielson [Nie80]. The second algorithm uses birational splines and is implemented according to Helmuth Spath [Spa91]. Interpolation in X and Z is performed using Akima's univariate interpolation scheme [Aki70, Aki91]. For the low-temperature opacites, tables from Kurucz [Kur91] and Alexander & Ferguson (1994, ApJ 437, 879) are available. For the Kurucz opacities tables for all X-values, as provided by the OPAL tables, are obtained by extrapolating the OPAL X-derivatives using Shepard's extrapolation scheme [She68]. The same method is used to estimate opacity values lying outside the given rectangular table domain.

Electron conduction (EC) according to Itoh et al. (1983, ApJ 273, 774) can be enabled for all tables which may become particular important for more massive stars.

2. Installation

If the machine (compiler), you are going to use OPINT on, can handle IEEE-floating point format, installation is very simple, using the provided binary table files from the package.

First look at the **Makefile** in the main-directory, and set up properly compiler-flages, ranlib, etc.

If your compiler can cope with the IEEE-floating point format (some compiler may require certain flags for it, e.g. for DEC/Alpha machines you would have to specify the compiler flag "-convert big_endian") then you only have to say "make lib" for (re-)building the whole library (./lib/libopint.a). After successfull compilation you are ready to use the routines in your programmes.

If, however, your compiler can't handle IEEE-FP-format, the binary opacity tables have to be computed as well. With "make tab92", "make tab95" and/or "make tab95a" the OPAL92+KURUCZ91, OPAL95+KURUCZ91 and/or OPAL95+ALEX94 tables are generated, respectively. (The Alexander tables are only available together with the OPAL95 tables). This procedure may take 10-30 minutes (depending on the machine's speed).

Saying "make" builds the library (./lib/libopint.a), test-programmes, as well as the procedures for generating the binary tables in the format as used by this package.

3. Usage

Three subroutines have to be called: 1.) **maceps**, 2.) **opinit** and 3.) **opintf** or **opintc** and/or **opints**. The first one (**maceps**) evaluates dynamically the relative machine precision, the second one (**opinit**) initializes the opacity-tables i.e., reads all tables. The third subroutine call (**opint**{**cfs**}) is the actual interpolation-routine (see below).

```
      implicit double precision (a-h,o-z)
c
c     test-driver program for opacity
c     interpolation subprograms
c     opinit       (opacity initialisation)
c     opintf       (opacity interpolation minimum norm)
c     opints       (opacity interpolation rational splines)
c
      character*79 tabnam
c
      data tabnam /'OPINTPATH_AX'/
      data iorder /4/
      data imode  /2/ ! initialize rat. splines & enables EC
c                     ! use imode /-2/ to disable EC
c
c     get relative machine precision
      call maceps(eps)
c
c     initialize opacity tables
      call opinit(eps,iorder,tabnam,imode)
c
```

Figure 1: Example how to initialize OPINT

The argument variable **iorder** in call **opinit** defines how many table-points should be used for the univariate interpolation in the X– and Z– domain respectively. This number defines the degree of the used univariate polynomial [Aki91]. Interpolation in Z is performed logarithmical and linear in X. With iorder=4 five tables will be used for the univariate interpolation in Z and X and will provide the most accurate interpolant (and the most expensive in computation time).

The initialisation subroutine **opinit** has the following arguments:

argument	type	inp/out	description
eps	double precision	input	used to evaluate certain machine-depend tolerances; evaluated from s/r maceps
iorder	integer	input	iorder+1 X and Z table-points will be used for the univariate interpolation in the X and Z domain, respectively $2 \leq$ iorder ≤ 4
tabnam	character*(80)	input	defines the filename, which opacity-tables should be used; take care of the sequence as given below in Table 3
imode	integer	input	selects algorithm(s) to be used; imode=0,1: only the minimum norm algorithm will be initialized (EC is enabled with imode=1) imode=2: only the birational splines are available (EC enabled) imode>2: both algorithm, the minimum norm and birational splines (+EC) are initialized (see also Table 4) imode≤0: a negative value of the particular imode number disables the contributions due to electron conduction

Table 2: argument description of routine opinit

The following table describes the fileformat of ”tabnam” (’OPINTPATH_AX’ in the above example [Figure 1]):

Entry #	filename	description
1	*your_path*/v9/opal95/opal95e.bin	Opacity values of OPAL-tables
2	*your_path*/v9/alex94/alex94.bin	Opacity values of Alexander-tables
3	*your_path*/v9/opal95/opal95e+alex94-pd.bin	Partial derivatives of OPAL- and Alexander- tables
4	*your_path*/v9/opal95/ival95.dat	used for determinig extrapolation domain of OPAL-tables

Table 3: format of ”tabnam” inputfile

Similar “tabnam”-files for the OPAL92/95 and Kurucz tables may be found in the main-directory with the name OPINTPATH_92 and OPINTPATH_95.

There are three interpolation-routines available, which have to be selected (initialized) properly by the argument variable **imode** in the previous call **opinit** (see also Table 2):

imode	interpolation-routine	description
1	opintf	minimum norm fast version **without** extrapolation-domain checking + EC
1	opintc	minimum norm version **with** extrapolation-domain checking + EC
2	opints	birational splines **with** extrapolation-domain checking + EC
≤0	Disables EC	Electron conduction (EC) is **disabled** with a negative value of the particular imode number (e.g., imode= -1 initializes either opintf or opintc and disables electron conduction

Table 4: available interpolation subroutine calls

The following listing is an excerpt of the programme doptesf.f:

```
     c
     c    interpolate to get opacity value opalg
     c    iexp=0
          ier=0
          call opintf(x,z,tlg,rlg,opalg,opr,opt,opx,opz,iexp,ier)
          if(ier.gt.0)write(6,'(a,i3)')
        +   'opint: ERROR in interpolation s/r, ier=',ier
     c    if(iexp.gt.0)print *,'nr. of extrapolated points: ',iexp
     c
```

Figure 2: Example how to use routine opintc

For the extrapolation domain checking of the OPAL-tables use **opintc** instead of **opintf** and uncomment:

```
     c    iexp=0
     c    if(iexp.gt.0)print *,'extrapolated points: ',iexp
```

Unfortunately the OPAL92-tables have no given values for the temperature range $4.65 < log_{10}(T) < 6.0$ and R-range $-0.5 < log_{10}(R) < 1.0$. Similar, the OPAL95 tables do not provide values for the region $7.2 < log_{10}(T) < 8.7$ and $-0.5 < log_{10}(R) < 1.0$. This area (here called the extrapolation domain) has been reconstructed by a proper extrapolation scheme using the univariate Akima interpolation scheme. The argument variable **iexp** indicates how many of these extrapolated table-points have been used in the interpolation process.

The argumentlist is the same for all three routines opintf, opintc and opints:

argument	type	inp/out	description
x	double precision	input	hydrogen mass fraction X $0. \leq x \leq 0.8$
z	double precision	input	mass fraction of heavy elements Z $0.0 \leq z \leq 0.04$ [0.1]
tlg	double precision	input	decade logarithm of temperature T $\{3.0\}3.32 \leq$tlg≤ 8.0 [8.7]
rlg	double precision	input	decade logarithm of $\frac{\rho}{T6^3}$; $\rho =$ density; $T6 = \frac{T}{10^6}$; [−8.] − 7. ≤rlg≤ 1.
opalg	double precision	output	decade logarithm of opacity κ
opr	double precision	output	partial derivative $\frac{\partial(log\kappa)}{\partial(log\rho)}_{T,X,Z}$
opt	double precision	output	partial derivative $\frac{\partial(log\kappa)}{\partial(logT)}_{\rho,X,Z}$
opx	double precision	output	partial derivative $\frac{\partial(log\kappa)}{\partial(logX)}_{\rho,T,Z}$
opz	double precision	output	partial derivative $\frac{\partial(log\kappa)}{\partial(logZ)}_{\rho,T,X}$
iexp	integer	output	opint[cs]: # of table points lying in the extrapolation domain opintf : 0
ier	integer	output	if ier = 1, the point lies outside the given table domain, and will be estimated by extrapolation using Shepard's method; otherwise ier=0

Table 5: argument description of routine opintf (opintc, opints). Squared brackets indicate values for OPAL95 and curly brackets for the Alexander94 low-T tables.

4. Linkage

After successful compilation of the OPINT-routines, the link editor under UNIX may be called as

```
$(F77) $(FFLAGS) your_programme   -Lyour_path/v9/lib   -lopint
```

5. Common-blocks

Following common blocks must not be altered between s/r call opinit and s/r opintf (opintc, opints):

```
c
c     pointers for the s/r opintd (used as argument
c     for s/r masubi)
      common /jpoint/ nti
      common /ipoint/ iali
c
c     partial derivatives and triangulation indices
      common /pderiv/ iwk(niwkl),wk(nwkl,ntab,nzva)
c
      common /opadat/ opa(nxir,nyir,ntab,nzva),
     +                rlg(nxir,ntab,nzva),
     +                tlg(nyir,ntab,nzva)
      common /valdat/ ivalo(nyir,ntab,nzva)
      common /tablex/ xtab(ntab),iorder
      common /tablez/ ztab(nzva)
      common /xyzdat/ xd(ndat,nzva),
     +                yd(ndat,nzva),
     +                zd(ndat,ntab,nzva)
      common /machin/ drelpr,toll,eps10
c
c     birational spline coefficients
      common /birasp/ pp,tls(nyif),
     +                ra(nxir,nyif,4,4,ntab,nzva)
c     table dimension for s/r opintcf and opints
      common /tabdim/ nzvai,ntabi,nxiri,nyiri,nyisi,iyifi,mdi,
     +                nti,iali
```

Figure 3: common-blocks used in the OPINT-package

6. File unit numbers

The following logical unit numbers are use in subroutine “opinit” in open-statements:

$lun = 31,\ 32,\ 33,$

and should therefore not be used in the calling routines.

References

[Aki70] H. Akima. A new method of interpolation and smooth curve fitting based on local procedures. *J. ACM*, 17(4):589–602, October 1970.

[Aki91] H. Akima. A method of univariate interpolation that has the accuracy of a third-degree polynomial. *ACM Trans. Math. Softw.*, 17(3):341–366, September 1991.

[Kur91] R. L. Kurucz. New opacity calculations. In L. Crivellari et al., editor, *Stellar Atmospheres: Beyond Classical Models*, pages 441–448, Netherlands, 1991. Kluwer Academic Publisher.

[MC89] L. B. Montefusco and G. Casciola. Algorithm 677 C ^{1} surface interpolation. *ACM Trans. Math. Softw.*, 15(4):365–374, December 1989.

[Nie80] G. M. Nielson. Minimum norm interpolation in triangles. *SIAM J. Numer. Anal.*, 17(1):44–62, February 1980.

[Nie83] G. M. Nielson. A method for interpolating scattered data based upon a minimum norm network. *J. Math. Comp.*, 40(161):253–271, January 1983.

[RI92] F.J. Rogers and C.A. Iglesias. Radiative atomic rosseland mean opacity tables. *Ap. J. Suppl.*, 79(2):507–568, April 1992.

[RI95] F.J. Rogers and C.A. Iglesias. In *A.S.P. Conf. Ser.*, volume 78, page 78. Brigham Young Univ., 1995.

[She68] D. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *Proceeding of the* 23^{rd} *National Conference*, pages 517–523, New York, 1968. ACM.

[Spa91] Helmuth Spath. *Zweidimenionae Spline-Interpolations-Algorithmen*. Reihe Datenverarbeitung. R.Oldenbourg Verlag, Munich Vienna, 1991.

Table of contents