```
      351 WL=WLBEG/RATIO
          DO 352 I=1,NPOINTS
          WL=WL*RATIO
          CT=HCKT*1.E7/WL
      352 ALINE(I)=ALINE(I)*FASTEX(CT)
      900 CONTINUE
          CALL EXIT
          END
```

7.9 Real Programs

The sample programs presented here represent only a few percent of the code. The complete opacity programs including equation of state, continuum opacity, hydrogen and helium lines, autoionizing lines, and lines merging into continua, together with Kurucz CD-ROM 1, the atomic line data, and Kurucz CD-ROM 15, the diatomic molecular line data, can be obtained from the author Kurucz@cfa.harvard.edu. These line lists date from 1975 to 1988 and are the data that have actually been used to compute opacities. In the next year I expect to completely redo and extend the line lists incorporating any new laboratory analyses and measurements.

References

Harris, D.L. (1949) Theoretical contours of absorption lines. Astrophysical Journal **109** (1982) 53-65

Kurucz, R.L. (1993a) ATLAS9 Stellar Atmosphere Programs and 2 km/s grid. Kurucz CD-ROM 13 (1993)

Kurucz, R.L. (1993b) SYNTHE Spectrum Synthesis Programs and Line Data. Kurucz CD-ROM 18 (1993)

Kurucz, R.L. (1994) Computation of opacities for diatomic molecules. To be published in IAU Colloquium 146: Molecular Opacities in the Stellar Environment (ed. by U.G. Jorgensen), Lecture Notes in Physics, Springer-Verlag, Berlin (1994)

Kurucz, R.L. and Avrett, E.H. (1981) Solar Spectrum Synthesis I. A Sample Atlas from 224 to 300 nm. Smithsonian Astrophysical Observatory Special Report No. 391 (1981)

Peytremann, E. (1970) personal communication.

This article was processed by the author using the TEX macro package from Springer-Verlag.

```
.             P=10.**PTAB(IP)
.             CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.             CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONTINUUM)
.             DO 202 I=1,NSPECIES
.           202 NUMPARTDOPRHO(I)=NUMPART(I)/DOPPLE(I)/RHO
.             DO 203 I=1,NPOINTS
.           203 ALINE(I)=0.
.             TNNN=(NHI+.42*NHEI+.85*NH2)*(T/10000.)**.3
.             REWIND 2
.             DO 350 ILINE=1,MLINES
.             READ(2)ICIEGGG
.       C     READ(2)IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.             CENTER=CONGF*NUMPARTDOPRHO(ISPECIES)
.             CUTOFF=ACONTINUUM(IPOINT)*.001
.             IF(CENTER.LT.CUTOFF)GO TO 350
.             ADAMP=(GAMRF+GAMSF*NELECTRON+GAMWF*TNNN)/DOPPLE(ISPECIES)
.             IF(ADAMP.LT..2)THEN
.             ALINE(IPOINT)=ALINE(IPOINT)+CENTER*(1.-1.128*ADAMP)
.             ELSE
.             ALINE(IPOINT)=ALINE(IPOINT)+CENTER*VOIGT(0.,ADAMP)
.             ENDIF
.       C     Profile inside 10 Doppler widths
.           319 N10DOP=10.*(DOPPLE(ISPECIES)*RESOLUTION)
.             JPOINT=IPOINT+MAXPROF
.             IF(ADAMP.LT..2)THEN
.             TABSTEP=VSTEPS/(DOPPLE(ISPECIES)*RESOLUTION)
.             TABI=1.5
.             DO 320 ISTEP=1,N10DOP
.             TABI=TABI+TABSTEP
.       C     Voigt approximation accurate only to ADAMP**2
.             PROFILE=CENTER*(HOTAB(IFIX(TABI))+ADAMP*H1TAB(IFIX(TABI)))
.             BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+PROFILE
.             BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+PROFILE
.             IF(PROFILE.LT.CUTOFF)GO TO 350
.           320 CONTINUE
.             ELSE
.             DVOIGT=1./DOPPLE(ISPECIES)/RESOLUTION
.             DO 321 ISTEP=1,N10DOP
.             PROFILE=CENTER*VOIGT(FLOAT(ISTEP)*DVOIGT,ADAMP)
.             BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+PROFILE
.             BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+PROFILE
.             IF(PROFILE.LT.CUTOFF)GO TO 350
.           321 CONTINUE
.             ENDIF
.       C     far wings
.             X=PROFILE*N10DOP**2
.             MAXSTEP=SQRT(X/CUTOFF)+1.
.             MAXSTEP=MIN(MAXSTEP,MAXPROF)
.             N1=N10DOP+1
.             DO 322 ISTEP=N1,MAXSTEP
.             BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+X/ISTEP**2
.           322 BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+X/ISTEP**2
.           350 CONTINUE
.       C     stimulated emission
```

```
.      1BLOCKSIZE=28000,RECL=7)
.       DO 900 IT=1,NT
.       T=10**TTAB(IT)
.       HCKT=6.62607E-27*2.99792458E10/1.380658E-16/T
.       DO 1 I=1,NSPECIES
.       DOPPLE(I)=
.      1 SQRT(2.*1.380658E-16*T/MASS(I)/1.66054E-24)/2.99792458E10
.       DO 1 IW=1,N10000
.     1 NUMMAX(I,IW)=0.
.C
.       DO 5 IP=1,NP
.       P=10.**PTAB(IP)
.       CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.       CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONT10000)
.       DO 2 IW=1,N10000
.       DO 2 I=1,NSPECIES
.       NUMM=NUMPART(I)/DOPPLE(I)/RHO/ACONT10000(IW)*1000.
.     2 NUMMAX(I,IW)=MAX(NUMMAX(I,IW),NUMM)
.     5 CONTINUE
.       REWIND 1
.       REWIND 2
.       MLINES=0
.       DO 6 ILINE=1,NLINES
.       READ(1)IIIIIII
.       IF(IWL.LT.IBEG)GO TO 6
.       IF(IWL.GT.IEND)GO TO 7
.       WL=DEXP(IWL*RATIOLOG)
.       ELO=TABLOG(IELO)
.       GF=TABLOG(IGFLOG)
.       GAMMAR=TABLOG(IGR)
.       GAMMAS=TABLOG(IGS)
.       GAMMAW=TABLOG(IGW)
.       ISPECIES=ICODE
.       IXWL=IWL*(RATIOLOG/RATIOLG)+.5D0
.       IPOINT=IXWL-IXWLBEG+1
.       FREQ=2.99792458E17/WL
.       CONGF=.01502*GF/FREQ
.       FRQ4PI=FREQ*12.5664
.       GAMRF=GAMMAR/FRQ4PI
.       GAMSF=GAMMAS/FRQ4PI
.       GAMWF=GAMMAW/FRQ4PI
.       I10000=IPOINT/10000+1
.       CENTER=CONGF*MAX(NUMMAX(ISPECIES,I10000),
.      1                 NUMMAX(ISPECIES,I10000+1))
.       IF(CENTER.LT.1.)GO TO 6
.       BOLT=FASTEX(ELO*HCKT)
.       IF(CENTER*BOLT.LT.1.)GO TO 6
.       CONGF=CONGF*BOLT
.       WRITE(2)ICIEGGG
.       MLINES=MLINES+1
.     6 CONTINUE
.     7 CONTINUE
.C
.       DO 900 IP=1,NP
```

```
.             PROGRAM FASTENOUGH
.             PARAMETER (NPOINTS=3507859,NSPECIES=594,NT=56,NP=21)
.             PARAMETER (RESOLUTION=500000.D0,NLINES=58000000,MAXPROF=20000)
.             PARAMETER (N10000=NPOINTS/10000+1)
.             REAL*4 ALINE(NPOINTS),ACONTINUUM(NPOINTS)
.             REAL*4 BUFFER(NPOINTS+2*MAXPROF)
.             EQUIVALENCE (ALINE(1),BUFFER(MAXPROF+1))
.             REAL*8 NUMPART(NSPECIES),NUMPARTDOPRHO(NSPECIES)
.             REAL*4 NUMMAX(NSPECIES,N10000),ACONT10000(N10000),NUMM
.             REAL*4 DOPPLE(NSPECIES),MASS(NSPECIES),TTAB(NT),PTAB(NP)
.             REAL*8 RATIOLG,RATIO,W,WL,WCEN,WLBEG,WLEND,RATIOLOG
.             COMMON /ICIEGGG/IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.             INTEGER*4 ICIEGGG(7)
.             EQUIVALENCE (IPOINT,ICIEGGG(1))
.             INTEGER*2 ICODE,IELO,IGFLOG,IGR,IGS,IGW
.             COMMON /IIIIIII/IWL,ICODE,IELO,IGFLOG,IGR,IGS,IGW
.             INTEGER*4 IIIIIII(4)
.             EQUIVALENCE (IIIIIII(1),IWL)
.             REAL*4 TABLOG(32768)
.             COMMON /EXTAB/EXTAB(1001),EXTABF(1001),E1TAB(2000)
.             DATA TTAB/3.32,3.34,3.36,3.38,3.40,3.42,3.44,3.46,3.48,3.50,
.            1 3.52,3.54,3.56,3.58,3.60,3.62,3.64,3.66,3.68,3.70,3.73,3.76,3.79,
.            2 3.82,3.85,3.88,3.91,3.94,3.97,4.00,4.05,4.10,4.15,4.20,4.25,4.30,
.            3 4.35,4.40,4.45,4.50,4.55,4.60,4.65,4.70,4.75,4.80,4.85,4.90,4.95,
.            4 5.00,5.05,5.10,5.15,5.20,5.25,5.30/
.             DATA PTAB/-2.0,-1.5,-1.0,-0.5,0.0,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,
.            1 4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.0/
.       C     put in real masses instead of dummy
.             DATA MASS/NSPECIES*33./
.       C
.             FASTEX(X)=EXTAB(IFIX(X)+1)*
.            1EXTABF(IFIX((X-FLOAT(IFIX(X)))*1000.+1.5))
.             DO 3456 I=1,1001
.             EXTAB(I)=EXP(-FLOAT(I-1))
.        3456 EXTABF(I)=EXP(-FLOAT(I-1)*.001)
.             DO 3457 I=1,32768
.        3457 TABLOG(I)=10.**((I-16384)*.001)
.       C     pretabulate Voigt function
.       C     200 steps per doppler width
.             VSTEPS=200.
.             CALL TABVOIGT(VSTEPS,2001)
.       C
.             RATIO=1.D0+1.D0/RESOLUTION
.             RATIOLG=LOG(RATIO)
.             WLBEG=8.97666
.             WLEND=10000.
.             RATIOLOG=DLOG(1.D0+1.D0/2000000.D0)
.             IBEG=DLOG(WLBEG)/RATIOLOG+.5
.             IEND=DLOG(WLEND)/RATIOLOG+.5
.             IXWLBEG=DLOG(WLBEG)/RATIOLG
.             IF(DEXP(IXWLBEG*RATIOLG).LT.WLBEG)IXWLBEG=IXWLBEG+1
.             OPEN(UNIT=1,STATUS='OLD',READONLY,FORM='UNFORMATTED',
.            1RECORDTYPE='FIXED',BLOCKSIZE=8000,RECL=4)
.             OPEN(UNIT=2,STATUS='NEW',FORM='UNFORMATTED',RECORDTYPE='FIXED',
```

```
      ENDIF
C     Profile inside 10 Doppler widths
  319 N10DOP=10.*(DOPPLE(ISPECIES)*RESOLUTION)
      JPOINT=IPOINT+MAXPROF
      IF(ADAMP.LT..2)THEN
      TABSTEP=VSTEPS/(DOPPLE(ISPECIES)*RESOLUTION)
      TABI=1.5
      DO 320 ISTEP=1,N10DOP
      TABI=TABI+TABSTEP
C     Voigt approximation accurate only to ADAMP**2
      PROFILE=CENTER*(HOTAB(IFIX(TABI))+ADAMP*H1TAB(IFIX(TABI)))
      BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+PROFILE
      BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+PROFILE
      IF(PROFILE.LT.CUTOFF)GO TO 350
  320 CONTINUE
      ELSE
      DVOIGT=1./DOPPLE(ISPECIES)/RESOLUTION
      DO 321 ISTEP=1,N10DOP
      PROFILE=CENTER*VOIGT(FLOAT(ISTEP)*DVOIGT,ADAMP)
      BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+PROFILE
      BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+PROFILE
      IF(PROFILE.LT.CUTOFF)GO TO 350
  321 CONTINUE
      ENDIF
C     far wings
      X=PROFILE*N10DOP**2
      MAXSTEP=SQRT(X/CUTOFF)+1.
      MAXSTEP=MIN(MAXSTEP,MAXPROF)
      N1=N10DOP+1
      DO 322 ISTEP=N1,MAXSTEP
      BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+X/ISTEP**2
  322 BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+X/ISTEP**2
  350 CONTINUE
C     stimulated emission
  351 WL=WLBEG/RATIO
      DO 352 I=1,NPOINTS
      WL=WL*RATIO
      CT=HCKT*1.E7/WL
  352 ALINE(I)=ALINE(I)*FASTEX(CT)
  900 CONTINUE
      CALL EXIT
      END
```

7.8 Program FASTENOUGH?

If there are 58,000,000 lines and each one takes 28 bytes, 1600 MB of disk storage are required. It is possible to pack the line data into 16 bytes per line so that only 900 MB of disk are required and the IO is reduced by a factor of 2. The tradeoff is that the unpacking requires CP time. I have written all my line data onto two CDROMs in this packed format.

```
.             T=10**TTAB(IT)
.             HCKT=6.62607E-27*2.99792458E10/1.380658E-16/T
.             DO 1 I=1,NSPECIES
.             DOPPLE(I)=
.            1 SQRT(2.*1.380658E-16*T/MASS(I)/1.66054E-24)/2.99792458E10
.             DO 1 IW=1,351
.           1 NUMMAX(I,IW)=0.
.       C
.             DO 5 IP=1,NP
.             P=10.**PTAB(IP)
.             CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.             CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONT10000)
.             DO 2 IW=1,N10000
.             DO 2 I=1,NSPECIES
.             NUMM=NUMPART(I)/DOPPLE(I)/RHO/ACONT10000(IW)*1000.
.           2 NUMMAX(I,IW)=MAX(NUMMAX(I,IW),NUMM)
.           5 CONTINUE
.             REWIND 1
.             REWIND 2
.             MLINES=0
.             DO 6 ILINE=1,NLINES
.             READ(1)ICIEGGG
.             I10000=IPOINT/10000+1
.             CENTER=CONGF*MAX(NUMMAX(ISPECIES,I10000),
.            1                 NUMMAX(ISPECIES,I10000+1))
.             IF(CENTER.LT.1.)GO TO 6
.             BOLT=FASTEX(ELO*HCKT)
.             IF(CENTER*BOLT.LT.1.)GO TO 6
.             CONGF=CONGF*BOLT
.             WRITE(2)ICIEGGG
.             MLINES=MLINES+1
.           6 CONTINUE
.       C
.             DO 900 IP=1,NP
.             P=10.**PTAB(IP)
.             CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.             CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONTINUUM)
.             DO 202 I=1,NSPECIES
.         202 NUMPARTDOPRHO(I)=NUMPART(I)/DOPPLE(I)/RHO
.             DO 203 I=1,NPOINTS
.         203 ALINE(I)=0.
.             TNNN=(NHI+.42*NHEI+.85*NH2)*(T/10000.)**.3
.             REWIND 2
.             DO 350 ILINE=1,MLINES
.             READ(2)ICIEGGG
.       C     READ(2)IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.             CENTER=CONGF*NUMPARTDOPRHO(ISPECIES)
.             CUTOFF=ACONTINUUM(IPOINT)*.001
.             IF(CENTER.LT.CUTOFF)GO TO 350
.             ADAMP=(GAMRF+GAMSF*NELECTRON+GAMWF*TNNN)/DOPPLE(ISPECIES)
.             IF(ADAMP.LT..2)THEN
.             ALINE(IPOINT)=ALINE(IPOINT)+CENTER*(1.-1.128*ADAMP)
.             ELSE
.             ALINE(IPOINT)=ALINE(IPOINT)+CENTER*VOIGT(0.,ADAMP)
```

table lookup twice as fast as by actually computing them. The arithmetic statement function FASTEX(X) replaces EXP(-X). This same replacement also works for stimulated emission.

At each temperature the line list is read NP times, once for each pressure. The exponential Boltzman factor is the same each time. If for each temperature we write a new line list that includes the Boltzman factor, it need be computed only once.

It is possible to determine the pressure for which the line strength relative to the continuum is a maximum. When the temperature changes, the number densities and continuum opacities are read in for all the pressures and the maxima are determined, but only for every 10000th wavelength point. Then the whole line list is read and compared to the minimum of the two continuum points nearest to each line. If a line passes this first test, it is tested again including the Boltzman factor. If it passes, it is written to a new file. These tests automatically eliminate all stages of ionization that are not populated.

```
      PROGRAM STILLFASTER
      PARAMETER (NPOINTS=3507859,NSPECIES=594,NT=56,NP=21)
      PARAMETER (N10000=NPOINTS/10000+1)
      PARAMETER (RESOLUTION=500000.D0,NLINES=58000000,MAXPROF=20000)
      REAL*4 ALINE(NPOINTS),ACONTINUUM(NPOINTS)
      REAL*4 BUFFER(NPOINTS+2*MAXPROF)
      EQUIVALENCE (ALINE(1),BUFFER(MAXPROF+1))
      REAL*8 NUMPART(NSPECIES),NUMPARTDOPRHO(NSPECIES)
      REAL*4 NUMMAX(NSPECIES,351),ACONT10000(N10000),NUMM
      REAL*4 DOPPLE(NSPECIES),MASS(NSPECIES),TTAB(NT),PTAB(NP)
      REAL*8 RATIOLG,RATIO,W,WL,WCEN
      COMMON /ICIEGGG/IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
      INTEGER*4 ICIEGGG(7)
      EQUIVALENCE (IPOINT,ICIEGGG(1))
      COMMON /EXTAB/EXTAB(1001),EXTABF(1001),E1TAB(2000)
C
      FASTEX(X)=EXTAB(IFIX(X)+1)*
     1EXTABF(IFIX((X-FLOAT(IFIX(X)))*1000.+1.5))
      DO 3456 I=1,1001
      EXTAB(I)=EXP(-FLOAT(I-1))
 3456 EXTABF(I)=EXP(-FLOAT(I-1)*.001)
C
      RATIO=1.D0+1.D0/RESOLUTION
      RATIOLG=LOG(RATIO)
      WLBEG=8.97666
      WLEND=10000.
C     pretabulate Voigt function
C     200 steps per doppler width
      VSTEPS=200.
      CALL TABVOIGT(VSTEPS,2001)
      OPEN(UNIT=1,STATUS='OLD',FORM='UNFORMATTED',RECORDTYPE='FIXED',
     1READONLY,BLOCKSIZE=28000,RECL=7)
      OPEN(UNIT=2,STATUS='NEW',FORM='UNFORMATTED',RECORDTYPE='FIXED',
     1BLOCKSIZE=28000,RECL=7)
      DO 900 IT=1,NT
```

```
.             IF(CENTER.LT.CUTOFF)GO TO 350
.             ADAMP=(GAMRF+GAMSF*NELECTRON+GAMWF*TNNN)/DOPPLE(ISPECIES)
.             IF(ADAMP.LT..2)THEN
.             ALINE(IPOINT)=ALINE(IPOINT)+CENTER*(1.-1.128*ADAMP)
.             ELSE
.             ALINE(IPOINT)=ALINE(IPOINT)+CENTER*VOIGT(0.,ADAMP)
.             ENDIF
.       C     Profile inside 10 Doppler widths
.         319 N10DOP=10.*(DOPPLE(ISPECIES)*RESOLUTION)
.             JPOINT=IPOINT+NAXPROF
.             IF(ADAMP.LT..2)THEN
.             TABSTEP=VSTEPS/(DOPPLE(ISPECIES)*RESOLUTION)
.             TABI=1.5
.             DO 320 ISTEP=1,N10DOP
.             TABI=TABI+TABSTEP
.       C     Voigt approximation accurate only to ADAMP**2
.             PROFILE=CENTER*(H0TAB(IFIX(TABI))+ADAMP*H1TAB(IFIX(TABI)))
.             BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+PROFILE
.             BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+PROFILE
.             IF(PROFILE.LT.CUTOFF)GO TO 350
.         320 CONTINUE
.             ELSE
.             DVOIGT=1./DOPPLE(ISPECIES)/RESOLUTION
.             DO 321 ISTEP=1,N10DOP
.             PROFILE=CENTER*VOIGT(FLOAT(ISTEP)*DVOIGT,ADAMP)
.             BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+PROFILE
.             BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+PROFILE
.             IF(PROFILE.LT.CUTOFF)GO TO 350
.         321 CONTINUE
.             ENDIF
.       C     far wings
.             X=PROFILE*N10DOP**2
.             MAXSTEP=SQRT(X/CUTOFF)+1.
.             MAXSTEP=MIN(MAXSTEP,MAXPROF)
.             N1=N10DOP+1
.             DO 322 ISTEP=N1,MAXSTEP
.             BUFFER(JPOINT+ISTEP)=BUFFER(JPOINT+ISTEP)+X/ISTEP**2
.         322 BUFFER(JPOINT-ISTEP)=BUFFER(JPOINT-ISTEP)+X/ISTEP**2
.         350 CONTINUE
.       C     stimulated emission
.             CT=HCKT*1.E7/WLBEG*RATIO
.             DO 352 I=1,NPOINTS
.         352 ALINE(I)=ALINE(I)*EXP(-CT/RATIO**IPOINT)
.          10 CONTINUE
.             CALL EXIT
.             END
```

7.7 Program STILLFASTER

The Boltzman factor does not have to be computed more accurately than 0.1%. On a VAX one can evaluate negative exponentials to that accuracy by

a maximum length of 20000 points for a line wing and I added buffers to each end of the spectrum. In FAST the logic allows lines outside the interval to have a wing running back into the interval. In FASTER those lines are thrown away. In practice, with the current line list, they do not matter.

```
.       PROGRAM FASTER
.       PARAMETER (NPOINTS=3507859,NSPECIES=594,NT=56,NP=21)
.       PARAMETER (RESOLUTION=500000.D0,NLINES=58000000,MAXPROF=20000)
.       REAL*4 ALINE(NPOINTS),ACONTINUUM(NPOINTS)
.       REAL*4 BUFFER(NPOINTS+2*MAXPROF)
.       EQUIVALENCE (ALINE(1),BUFFER(MAXPROF+1))
.       REAL*8 NUMPART(NSPECIES),NUMPARTDOPRHO(NSPECIES)
.       REAL*4 DOPPLE(NSPECIES),MASS(NSPECIES),TTAB(NT),PTAB(NP)
.       REAL*8 RATIOLG,RATIO,W,WL,WCEN,WLBEG,WLEND
.       COMMON /ICIEGGG/IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.       INTEGER*4 ICIEGGG(7)
.       EQUIVALENCE (IPOINT,ICIEGGG(1))
.       RATIO=1.D0+1.D0/RESOLUTION
.       RATIOLG=LOG(RATIO)
.       WLBEG=8.97666
.       WLEND=10000.
.C      pretabulate Voigt function
.C      200 steps per doppler width
.       VSTEPS=200.
.       CALL TABVOIGT(VSTEPS,2001)
.       OPEN(UNIT=1,STATUS='NEW',FORM='UNFORMATTED',RECORDTYPE='FIXED',
.      1READONLY,BLOCKSIZE=32000,RECL=8)
.       DO 10 IT=1,NT
.       T=10**TTAB(IT)
.       DO 1 ISPECIES=1,NSPECIES
.     1 DOPPLE(I)=
.      1 SQRT(2.*1.380658E-16*T/MASS(I)/1.66054E-24)/2.99792458E10
.       HCKT=6.62607E-27*2.99792458E10/1.380658E-16/T
.       DO 10 IP=1,NP
.C or   DO 10 IRHO=1,NRHO
.       P=10.**PTAB(IP)
.C      get EOS and continuous opacity from Hoeflich (this volume) or
.C      Werner (this volume) or Kurucz (1993ab) or elsewhere
.       CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.       CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONTINUUM)
.       DO 2 ISPECIES=1,NSPECIES
.     2 NUMPARTDOPRHO(ISPECIES)=NUMPART(ISPECIES)/DOPPLE(ISPECIES)/RHO
.       DO 3 I=1,NPOINTS
.     3 ALINE(I)=0.
.C      see Kurucz and Avrett (1981)
.       TNNN=(NHI+.42*NHEI+.85*NH2)*(T/10000.)**.3
.       REWIND 1
.       DO 350 ILINE=1,NLINES
.       READ(1)ICIEGGG
.C      READ(1)IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.       CENTER=CONGF*NUMPARTDOPRHO(ISPECIES)
.       CUTOFF=ACONTINUUM(IPOINT)*.001
.       IF(CENTER.LT.CUTOFF)GO TO 350
.       CENTER=CENTER*EXP(-ELO*HCKT)
```

```
      IF(L.GT.LL+1.OR.L.EQ.3)GO TO 21
      CBAC=CFOR
      BBAC=BFOR
      ABAC=AFOR
      IF(L.EQ.NOLD)GO TO 22
      GO TO 25
   21 L2=L-2
      D=(FOLD(L1)-FOLD(L2))/(XOLD(L1)-XOLD(L2))
      CBAC=FOLD(L)/((XOLD(L)-XOLD(L1))*(XOLD(L)-XOLD(L2)))+
     1(FOLD(L2)/(XOLD(L)-XOLD(L2))-FOLD(L1)/(XOLD(L)-XOLD(L1)))/
     2(XOLD(L1)-XOLD(L2))
      BBAC=D-(XOLD(L1)+XOLD(L2))*CBAC
      ABAC=FOLD(L2)-XOLD(L2)*D+XOLD(L1)*XOLD(L2)*CBAC
      IF(L.LT.NOLD)GO TO 25
   22 C=CBAC
      B=BBAC
      A=ABAC
      LL=L
      GO TO 50
   25 D=(FOLD(L)-FOLD(L1))/(XOLD(L)-XOLD(L1))
      CFOR=FOLD(L+1)/((XOLD(L+1)-XOLD(L))*(XOLD(L+1)-XOLD(L1)))+
     1(FOLD(L1)/(XOLD(L+1)-XOLD(L1))-FOLD(L)/(XOLD(L+1)-XOLD(L)))/
     2(XOLD(L)-XOLD(L1))
      BFOR=D-(XOLD(L)+XOLD(L1))*CFOR
      AFOR=FOLD(L1)-XOLD(L1)*D+XOLD(L)*XOLD(L1)*CFOR
      WT=0.
      IF(ABS(CFOR).NE.0.)WT=ABS(CFOR)/(ABS(CFOR)+ABS(CBAC))
      A=AFOR+WT*(ABAC-AFOR)
      B=BFOR+WT*(BBAC-BFOR)
      C=CFOR+WT*(CBAC-CFOR)
      LL=L
      GO TO 50
   30 IF(L.EQ.LL)GO TO 50
      L=AMINO(NOLD,L)
      C=0.
      B=(FOLD(L)-FOLD(L-1))/(XOLD(L)-XOLD(L-1))
      A=FOLD(L)-XOLD(L)*B
      LL=L
   50 FNEW(K)=A+(B+C*XNEW(K))*XNEW(K)
      RETURN
      END
```

7.6 Program FASTER

Much of the logic in program FAST deals with endpoints. In computing small intervals of spectrum the chance of having a line wing run to an endpoint is not small. In this case however the whole spectrum is computed at once so the chance of reaching an endpoint is small. It is much more efficient to append buffers to the beginning and end of the spectrum and to allow line wings to run off the ends. In this example I arbitrarily chose

```
.      8 .0081722,.0077885,.0074314,.0070985,.0067875,.0064967,.0062243,
.      9 .0059688,.0057287,.0055030,.0052903,.0050898,.0049006,.0047217,
.      T .0045526,.0043924,.0042405,.0040964,.0039595/
.      C     PRETABULATE VOIGT FUNCTION
.      C     100 STEPS PER DOPPLER WIDTH GIVES 2 PER CENT ACCURACY
.            DO 1 I=1,N
.          1 HOTAB(I)=FLOAT(I-1)/VSTEPS
.            CALL MAP1(TABVI,TABH1,81,HOTAB,H1TAB,N)
.            DO 2 I=1,N
.            VV=(FLOAT(I-1)/VSTEPS)**2
.            HOTAB(I)=EXP(-VV)
.          2 H2TAB(I)=HOTAB(I)-(VV+VV)*HOTAB(I)
.            RETURN
.            END
.            FUNCTION VOIGT(V,A)
.      C     FAST VOIGT
.            COMMON /H1TAB/HOTAB(2001),H1TAB(2001),H2TAB(2001)
.            IV=V*200.+1.5
.            IF(A.LT..2)GO TO 10
.            IF(A.GT.1.4)GO TO 2
.            IF(A+V.GT.3.2)GO TO 2
.            VV=V*V
.            HH1=H1TAB(IV)+HOTAB(IV)*1.12838
.            HH2=H2TAB(IV)+HH1*1.12838-HOTAB(IV)
.            HH3=(1.-H2TAB(IV))*.37613-HH1*.66667*VV+HH2*1.12838
.            HH4=(3.*HH3-HH1)*.37613+HOTAB(IV)*.66667*VV*VV
.            VOIGT=((((HH4*A+HH3)*A+HH2)*A+HH1)*A+HOTAB(IV))*
.           1      (((-.122727278*A+.532770573)*A-.96284325)*A+.979895032)
.            RETURN
.          2 AA=A*A
.            VV=V*V
.            U=(AA+VV)*1.4142
.            UU=U*U
.            VOIGT=
.           A  ((((AA-10.*VV)*AA*3.+15.*VV*VV)/UU+3.*VV-AA)/UU+1.)*A*.79788/U
.            RETURN
.         10 IF(V.GT.10.)GO TO 12
.         11 VOIGT=(H2TAB(IV)*A+H1TAB(IV))*A+HOTAB(IV)
.            RETURN
.         12 VOIGT=.5642*A/V**2
.            RETURN
.            END
.            SUBROUTINE MAP1(XOLD,FOLD,NOLD,XNEW,FNEW,NNEW)
.            DIMENSION XOLD(*),FOLD(*),XNEW(*),FNEW(*)
.            L=2
.            LL=0
.            DO 50 K=1,NNEW
.         10 IF(XNEW(K).LT.XOLD(L))GO TO 20
.            L=L+1
.            IF(L.GT.NOLD)GO TO 30
.            GO TO 10
.         20 IF(L.EQ.LL)GO TO 50
.            IF(L.EQ.2)GO TO 30
.            L1=L-1
```

```
      324 ALINE(IPOINT+ISTEP)=ALINE(IPOINT+ISTEP)+PROFILE(ISTEP)
          IF(IPOINT.LE.1)GO TO 350
C     blue wing
      325 MAXBLUE=MINO(IPOINT-1,NSTEP)
          MINBLUE=MAXO(1,IPOINT-NPOINTS)
          DO 326 ISTEP=MINBLUE,MAXBLUE
      326 ALINE(IPOINT-ISTEP)=ALINE(IPOINT-ISTEP)+PROFILE(ISTEP)
      350 CONTINUE
C     stimulated emission
          CT=HCKT*1.E7/WLBEG*RATIO
          DO 352 I=1,NPOINTS
      352 ALINE(I)=ALINE(I)*EXP(-CT/RATIO**I)
       10 CONTINUE
          CALL EXIT
          END
```

7.5 Subroutine VOIGT

The Voigt function need be computed no more accurately than the physical accuracy. One percent errors are probably good enough and 0.1% errors are unobservable. This allows all sorts of numerical approximations and even table lookup for evaluating the function. Here, for small damping parameter A, I follow Harris (1949) and expand the Voigt function in a power series in A and then pretabulate the coefficients as functions of doppler width V, at 200 points per doppler width. The fits for other ranges were taken from a subroutine by Peytremann (1970).

```
C     SUBROUTINE VOIGT
C
C     pretabulate Voigt function
C     200 steps per doppler width
C     VSTEPS=200.
C     CALL TABVOIGT(VSTEPS,2001)
C
      SUBROUTINE TABVOIGT(VSTEPS,N)
      COMMON /H1TAB/HOTAB(2001),H1TAB(2001),H2TAB(2001)
      DIMENSION TABVI(81),TABH1(81)
      DATA TABVI/0.,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.,1.1,1.2,1.3,1.4,1.5,
     11.6,1.7,1.8,1.9,2.,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.,3.1,3.2,
     2 3.3,3.4,3.5,3.6,3.7,3.8,3.9,4.0,4.2,4.4,4.6,4.8,5.0,5.2,5.4,5.6,
     3 5.8,6.0,6.2,6.4,6.6,6.8,7.0,7.2,7.4,7.6,7.8,8.0,8.2,8.4,8.6,8.8,
     4 9.0,9.2,9.4,9.6,9.8,10.0,10.2,10.4,10.6,10.8,11.0,11.2,11.4,11.6,
     5 11.8,12.0/
      DATA TABH1/-1.12838,-1.10596,-1.04048,-.93703,-.80346,-.64945,
     1-.48552,-.32192,-.16772,-.03012,.08594,.17789,.24537,.28981,
     2.31394,.32130,.31573,.30094,.28027,.25648,.231726,.207528,.184882,
     3.164341,.146128,.130236,.116515,.104739,.094653,.086005,.078565,
     4 .072129,.066526,.061615,.057281,.053430,.049988,.046894,.044098,
     5 .041561,.039250,.035195,.031762,.028824,.026288,.024081,.022146,
     6 .020441,.018929,.017582,.016375,.015291,.014312,.013426,.012620,
     7 .0118860,.0112145,.0105990,.0100332,.0095119,.0090306,.0085852,
```

```
.         2 NUMPARTDOPRHO(ISPECIES)=NUMPART(ISPECIES)/DOPPLE(ISPECIES)/RHO
.           DO 3 I=1,NPOINTS
.         3 ALINE(I)=0.
.       C   see Kurucz and Avrett (1981)
.           TNNN=(NHI+.42*NHEI+.85*NH2)*(T/10000.)**.3
.           REWIND 1
.           MLINES=0
.           DO 350 ILINE=1,NLINES
.           READ(1)ICIEGGG
.       C   READ(1)IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.           CENTER=CONGF*NUMPARTDOPRHO(ISPECIES)
.           CUTOFF=ACONTINUUM(IPOINT)*.001
.           IF(CENTER.LT.CUTOFF)GO TO 350
.           CENTER=CENTER*EXP(-ELO*HCKT)
.           IF(CENTER.LT.CUTOFF)GO TO 350
.           ADAMP=(GAMRF+GAMSF*NELECTRON+GAMWF*TNNN)/DOPPLE(ISPECIES)
.           MLINES=MLINES+1
.           IF(ADAMP.LT..2)THEN
.           ALINE(IPOINT)=ALINE(IPOINT)+CENTER*(1.-1.128*ADAMP)
.           ELSE
.           ALINE(IPOINT)=ALINE(IPOINT)+CENTER*VOIGT(0.,ADAMP)
.           ENDIF
.       C   profile inside 10 doppler widths
.           N10DOP=10.*(DOPPLE(ISPECIES)*RESOLUTION)
.           IF(ADAMP.LT..2)THEN
.           TABSTEP=VSTEPS/(DOPPLE(ISPECIES)*RESOLUTION)
.           TABI=1.5
.           DO 320 NSTEP=1,N10DOP
.           TABI=TABI+TABSTEP
.       C   Voigt approximation accurate only to ADAMP**2
.           PROFILE(NSTEP)=CENTER*(HOTAB(IFIX(TABI))+ADAMP*H1TAB(IFIX(TABI)))
.           IF(PROFILE(NSTEP).LT.CUTOFF)GO TO 323
.       320 CONTINUE
.           ELSE
.           DVOIGT=1./DOPPLE(ISPECIES)/RESOLU
.           DO 321 NSTEP=1,N10DOP
.           PROFILE(NSTEP)=CENTER*VOIGT(FLOAT(NSTEP)*DVOIGT,ADAMP)
.           IF(PROFILE(NSTEP).LT.CUTOFF)GO TO 323
.       321 CONTINUE
.           ENDIF
.       C   far wings
.           X=PROFILE(N10DOP)*FLOAT(N10DOP)**2
.           MAXSTEP=SQRT(X/CUTOFF)+1.
.           MAXSTEP=MIN(MAXSTEP,MAXPROF)
.           N1=N10DOP+1
.           DO 322 NSTEP=N1,MAXSTEP
.       322 PROFILE(NSTEP)=X/FLOAT(NSTEP)**2
.           NSTEP=MAXSTEP
.       323 IF(IPOINT+NSTEP.LT.1.OR.IPOINT-NSTEP.GT.NPOINTS)GO TO 350
.           IF(IPOINT.GE.NPOINTS)GO TO 325
.       C   red wing
.           MAXRED=MIN0(NPOINTS-IPOINT,NSTEP)
.           MINRED=MAX0(1,1-IPOINT)
.           DO 324 ISTEP=MINRED,MAXRED
```

determined and affect the line width. In that case the resolution must be increased by an order of magnitude or approximations are not valid. In the example discussed here, and in all my published work, hyperfine and isotopic splitting are not included. The approximation I use is to symmetrize the Voigt profile by moving the line center to the nearest point. Then only one half the profile need be evaluated. The maximum wavelength error is thus one half the resolution, or 10^{-6} in this case, and the error can be 0. The point spacing is uniform in doppler width. In the input line list the real*8 wavelength can be replaced by an integer*4 point number thereby saving 4 bytes per line. Full accuracy is also not needed for the lower energy level. It can be stored as real*4 instead of real*8, saving another 4 bytes per line.

The logic for the line profile treatment is taken from my spectrum synthesis program SYNTHE (Kurucz 1993b). It also takes advantage of simple approximations for the Voigt function for small A and for large V to avoid making the function call.

```
.       PROGRAM FAST
.       PARAMETER (NPOINTS=3507859,NSPECIES=594,NT=56,NP=21)
.       PARAMETER (RESOLUTION=500000.D0,NLINES=58000000)
.       REAL*4 ALINE(NPOINTS),ACONTINUUM(NPOINTS),PROFILE(20000)
.       REAL*8 NUMPART(NSPECIES),NUMPARTDOPRHO(NSPECIES)
.       REAL*4 DOPPLE(NSPECIES),MASS(NSPECIES),TTAB(NT),PTAB(NP)
.       REAL*8 RATIOLG,RATIO,W,WL,WCEN,WLBEG,WLEND
.       COMMON /H1TAB/HOTAB(2001),H1TAB(2001),H2TAB(2001)
.       COMMON /ICIEGGG/IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
.       INTEGER*4 ICIEGGG(7)
.       EQUIVALENCE (IPOINT,ICIEGGG(1))
.       RATIO=1.D0+1.D0/RESOLUTION
.       RATIOLG=LOG(RATIO)
.       WLBEG=8.97666
.       WLEND=10000.
.C      pretabulate Voigt function
.C      200 steps per doppler width
.       VSTEPS=200.
.       CALL TABVOIGT(VSTEPS,2001)
.       OPEN(UNIT=1,STATUS='OLD',FORM='UNFORMATTED',RECORDTYPE='FIXED',
.      1READONLY,BLOCKSIZE=28000,RECL=7)
.       DO 10 IT=1,NT
.       T=10**TTAB(IT)
.       DO 1 ISPECIES=1,NSPECIES
.     1 DOPPLE(I)=
.      1 SQRT(2.*1.380658E-16*T/MASS(I)/1.66054E-24)/2.99792458E10
.       HCKT=6.62607E-27*2.99792458E10/1.380658E-16/T
.       DO 10 IP=1,NP
.C or   DO 10 IRHO=1,NRHO
.       P=10.**PTAB(IP)
.C      get EOS and continuous opacity from Hoeflich (this volume) or
.C      Werner (this volume) or Kurucz (1993ab) or elsewhere
.       CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.       CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONTINUUM)
.       DO 2 ISPECIES=1,NSPECIES
```

with 1000 records per block. In this presentation I am ignoring double buffering and operating-system-dependent IO primitives. It is possible to set up the opacity program so that it computes with one block of lines while reading in the next block.

```
      PROGRAM REWRITELINES
      PARAMETER (NPOINTS=3507859)
      PARAMETER (RESOLUTION=500000.D0,NLINES=58000000)
      REAL*8 RATIOLG,RATIO,W,WL,WCEN,ELO8
      REAL*4 ELO
      RATIO=1.D0+1.D0/RESOLUTION
      RATIOLG=LOG(RATIO)
      WLBEG=8.97666
      WLEND=10000.
C     pi***2/m/c
      CONSTANT=.02654
      MLINES=0
      OPEN(UNIT=2,STATUS='NEW',FORM='UNFORMATTED',RECORDTYPE='FIXED',
     1BLOCKSIZE=28000,RECL=7)
      DO 9 ILINE=1,NLINES
      READ(1)WL,ELO,GF,ISPECIES,GAMMARAD,GAMMASTARK,GAMMAWAALS
      IPOINT=DLOG(WL-WLBEG)/RATIOLG
      IF(IPOINT.LT.1)GO TO 9
      IF(IPOINT.GT.NPOINTS)GO TO 10
      MLINES=MLINES+1
      FREQ=2.99792458E17/WL
      CONGF=CONSTANT*GF/FREQ/1.77245385
      GAMRF=GAMMARAD/12.5664/FREQ
      GAMSF=GAMMASTARK/12.5664/FREQ
      GAMWF=GAMMAWAALS/12.5664/FREQ
      ELO=ELO8
      WRITE(2)IPOINT,CONGF,ISPECIES,ELO,GAMRF,GAMSF,GAMWF
    9 CONTINUE
   10 PRINT *,MLINES
      CALL EXIT
      END
```

7.4 Program FAST

There are limits to the physical accuracy. Very few gf values are known more accurately than 1%. Few are known to better than 10%. Most are known only to a factor of 2. The accuracy of the damping constants is between 10% and 1000%. Only a fraction between 10 and 50% of the energy levels have been accurately determined in the laboratory. Those and wavelengths derived from them can be accurate to 10^{-8}. Other levels must be predicted with accuracies on the order of 10^{-4} and with wavelength accuracy 10^{-3} to 10^{-4}. The actual position of most of the lines used is uncertain. As long as we are computing at high resolution, we are allowed some freedom to adjust the wavelengths to simplify the calculation. An exception would be treating hyperfine and isotopic splitting where the splittings can be very accurately

```
.            READ(1)WL,ELO,GF,ISPECIES,GAMMARAD,GAMMASTARK,GAMMAWAALS
.            IF(WL.LT.WLBEG)GO TO 9
.            IF(WL.GT.WLEND)GO TO 10
.            FREQ=2.99792458E17/WL
.            DOPFREQ=FREQ*DOPPLE(ISPECIES)
.            CENTER=CONSTANT*GF*EXP(-ELO*HCKT)*NUMPART(ISPECIES)/RHO/DOPFREQ/
.           1 1.77245385*(1.-EXP(-FREQ*HKT))
.            IPOINT=DLOG(WL-WLBEG)/RATIOLG
.            CUTOFF=ACONTINUUM(IPOINT)
.            IF(CENTER.LT.CUTOFF)GO TO 9
.            ADAMP=(GAMMARAD+GAMMASTARK*WELECTRON+GAMMAWAALS*TWNN)/12.4556/
.           1 DOPFREQ
.            DOPWL=WL*DOPPLE(ISPECIES)
.            WCEN=WLBEG*RATIO**(IPOINT-1)
.            W=WCEN
.            DO 5 I=IPOINT+1,NPOINTS
.            W=W*RATIO
.      C     use any Voigt routine that is normalized to 1 for VOIGT(0.,0.)
.            PROFILE=CENTER*VOIGT((W-WL)/DOPWL,ADAMP)
.            ALINE(I)=ALINE(I)+PROFILE
.            IF(PROFILE.LT.CUTOFF)GO TO 6
.          5 CONTINUE
.          6 W=WCEN*RATIO
.            DO 7 I=IPOINT,1,-1
.            W=W/RATIO
.            PROFILE=CENTER*VOIGT((WL-W)/DOPWL,ADAMP)
.            ALINE(I)=ALINE(I)+PROFILE
.            IF(PROFILE.LT.CUTOFF)GO TO 9
.          7 CONTINUE
.          9 CONTINUE
.      C
.      C     ALINE now contains the line opacity spectrum for T,P  To get the.
.      C     total opacity add together ALINE and ACONTINUUM. Either write out the
.      C     whole spectrum, or statistically analyze it to determine distribution
.      C     functions, or compute means.  To evaluate the  Rosseland mean it may
.      C     be necessary to add an extension on the red or blue end depending on
.      C     the temperature.  In practice I include only continuous opacity in
.      C     such extensions.
.      C
.      C     To introduce a microturbulent velocity convolve the 0 km/s spectrum
.      C     with a gaussian.
.      C
.         10 CONTINUE
.            CALL EXIT
.            END
```

7.3 Program REWRITELINES

This program simply multiplies the gf value and all the damping constants by the constant factors. This saves NT x NP x a few operations per line. To make IO more efficient it also writes the data in fixed length 28 byte records

is greater than say 10000K. The highly ionized atoms are not read if the temperature is less than say 10000K.

This program will run very slowly and produce an opacity spectrum that is numerically accurate to five or six decimal places. By factoring the calculation into temperature dependent and temperature independent parts, by minimizing the number of exponential and Voigt function calls, and by preprocessing the line data, we can make it run considerably faster. By considering the physical accuracy of the problem, we can make approximations that reduce the numerical accuracy to the physical accuracy and greatly reduce the running time. Altogether an improvement by a factor of 1000 is possible.

```
.       PROGRAM SLOW
.       PARAMETER (NPOINTS=3507859,NSPECIES=594,NT=56,NP=21)
.       PARAMETER (RESOLUTION=500000.D0,NLINES=58000000)
.       REAL*4 ALINE(NPOINTS),ACONTINUUM(NPOINTS)
.       REAL*8 NUMPART(NSPECIES)
.       REAL*4 DOPPLE(NSPECIES),MASS(NSPECIES),TTAB(NT),PTAB(NP)
.       REAL*8 RATIOLG,RATIO,W,WL,WCEN,ELO,WLBEG,WLEND
.       DATA TTAB/3.32,3.34,3.36,3.38,3.40,3.42,3.44,3.46,3.48,3.50,
.      1 3.52,3.54,3.56,3.58,3.60,3.62,3.64,3.66,3.68,3.70,3.73,3.76,3.79,
.      2 3.82,3.85,3.88,3.91,3.94,3.97,4.00,4.05,4.10,4.15,4.20,4.25,4.30,
.      3 4.35,4.40,4.45,4.50,4.55,4.60,4.65,4.70,4.75,4.80,4.85,4.90,4.95,
.      4 5.00,5.05,5.10,5.15,5.20,5.25,5.30/
.       DATA PTAB/-2.0,-1.5,-1.0,-0.5,0.0,0.5,1.0,1.5,2.0,2.5,3.0,3.5,4.0,
.      1 4.5,5.0,5.5,6.0,6.5,7.0,7.5,8.0/
.C      put in real masses instead of dummy
.       DATA MASS/NSPECIES*33./
.       RATIO=1.D0+1.D0/RESOLUTION
.       RATIOLG=LOG(RATIO)
.       WLBEG=8.97666
.       WLEND=10000.
.C      pi*e**2/m/c
.       CONSTANT=.02654
.       DO 10 IT=1,NT
.       T=10**TTAB(IT)
.       DO 1 I=1,NSPECIES
.     1 DOPPLE(I)=
.      1 SQRT(2.*1.380658E-16*T/MASS(I)/1.66054E-24)/2.99792458E10
.       HCKT=6.62607E-27*2.99792458E10/1.380658E-16/T
.       DO 10 IP=1,NP
.C or   DO 10 IRHO=1,NRHO  density instead of pressure as the variable
.       P=10**PTAB(IP)
.C      get EOS and continuous opacity from Hoeflich (this volume) or
.C      Werner (this volume) or Kurucz (1993ab) or elsewhere
.       CALL EQSTATE(T,P,RHO,NUMPART,NELECTRON,NHI)
.       CALL CONTINUUM(T,P,RHO,NUMPART,NELECTRON,NHI,ACONTINUUM)
.       DO 3 I=1,NPOINTS
.     3 ALINE(I)=0.
.C      see Kurucz and Avrett (1981) for the damping treatment
.       TNNN=(NHI+.42*NHEI+.85*NH2)*(T/10000.)**.3
.       REWIND 1
.       DO 9 ILINE=1,NLINES
```

Chapter 7

Rapid Calculation of Line Opacity

Robert L. Kurucz

7.1 Introduction

I have developed numerical and programming techniques for rapidly computing the summed opacity of millions of lines without making compromises in the physics. These techniques have been used both for tabulating opacity for stellar atmospheres and envelopes and for computing stellar and terrestrial atmospheric spectra in high precision (Kurucz 1994). Successively faster sample programs are presented to illustrate the methodology.

7.2 Program SLOW

I wrote SLOW in a few minutes as a generic code for computing line opacity assuming that all lines have Voigt profiles. I have set the parameters to be the ones I actually use in my distribution function opacity calculations, namely, wavelength range 8.97666 to 10000 nm, resolution 500000 = 0.6 km/s, 56 temperatures from 2089K to 199526K, 21 pressures from log P = -2 to +8, and approximately 58 million lines. My line data are not adequate at the lowest temperatures because only diatomic molecules are included, and they are not adequate at high temperatures because they do not go beyond the tenth stage of ionization (charge +9). At the highest temperatures it would be possible to run at lower resolution more appropriate to the thermal width. At the lowest temperatures the resolution should be doubled. When I am able to include triatomic molecules, I may make the resolution a function of temperature.

Making a naive estimate, the program must read 58 million lines 1176 times, or 68 billion lines total. In practice the line data are divided up by temperature. The diatomic molecules are not read if the temperature