

CINDY A. COHN, ESQ.; SBN 145997  
McGLASHAN & SARRAIL  
Professional Corporation  
177 Bovet Road, Sixth Floor  
San Mateo, CA 94402  
Tel: (415) 341-2585  
Fax: (415) 341-1395

LEE TIEN, ESQ.; SBN 148216  
1452 Curtis Street  
Berkeley, CA 94702  
Tel: (510) 525-0817

Attorneys for Plaintiff  
Daniel J. Bernstein

IN THE UNITED STATES DISTRICT COURT  
FOR THE NORTHERN DISTRICT OF CALIFORNIA

DANIEL J. BERNSTEIN )  
) C 95-00582 MHP  
Plaintiff, )  
) DECLARATION OF  
v. ) DR. PAUL GINSPARG  
) )  
UNITED STATES DEPARTMENT OF )  
STATE et al., ) )  
Defendants. )  
)  
)

I, Dr. Paul Ginsparg, hereby declare:

1. I am currently a staff physicist at the Los Alamos National Laboratory, where I have been since 1990. Prior to this, from 1984 until 1990, I was a professor of Physics at Harvard University. I have been actively involved in the science of physics since I graduated summa cum laude - A.B. Physics from Harvard University in 1977, and earned a Ph.D. - Physics from Cornell University in 1981.
2. In this declaration, I speak on behalf of myself and not my employer, the Los Alamos National Laboratory.
3. Since 1991 I have been actively involved in the development of several online systems that allow for dissemination, review, analysis and testing of physics over the Internet. The systems, in essence, create giant online databases of physics ideas which can be accessed by physicists around the world. These systems are entirely automated (including submission process and indexing of titles/authors/abstracts), allow access via e-mail, anonymous file transfer and the World Wide Web.
4. In addition to the systems I have created, many physicists utilize Internet discussion groups. Some of these discussion groups are part of the Internet system known as Usenet. Usenet includes many groups dedicated to the discussion of sciences, including physics. The discussion group at issue in this case, sci.crypt, is one Usenet discussion group. Physics-related newsgroups include sci.physics, sci.physics.fusion and sci.physics.research.
5. These systems and discussion groups are a fundamental part of the development of science. They are the natural extension of, and I believe the successor to, print publication of ideas.
6. Publication of ideas is vital to the scientific method. Scientists toss their ideas, and the implementation scheme for these ideas, out into the "marketplace of ideas" and allow their peers and others to evaluate them,

test them, and discuss them. This process weeds out bad ideas from the good ones.

7. Scientists are increasingly using the Internet to publish new ideas and scientific findings. The Internet allows scientists scattered across the world to share, evaluate, and discuss their ideas more quickly, including those ideas expressed or contained in computer code. To the scientific community, publishing something on the Internet should be protected from government regulation in the same way that publishing something on paper is -- the difference in media should make no difference in the type of legal protection.

8. The electronic medium fostered by the Internet allows scientists to do much more than does the comparatively feeble print medium, since it facilitates a more generalized format for distribution of research results, including attainment of raw data and in some cases all necessary tools to repeat theoretical or experimental analysis of it.

9. For computer scientists, physicists, mathematicians, economists, or anyone whose ideas include or involve computer codes, publication of their ideas includes publication of the computer code.

10. The availability of computer code, mathematical equations and written language allows scientists to choose the most precise means to express their theories, problems, or models.

11. Because greater precision and clarity is important in scientific discovery, the scientific community and subsequently the public benefit from the publication of computer code.

12. One Internet site allowing scientists to communicate and exchange ideas is the Los Alamos E-Print Archives, a project of mine. The E-Print Archives are located at the Internet address, <http://xxx.lanl.gov>. Allowing researchers to transfer unedited copies of their research papers, the archives contain research papers from more than ten physical sciences and mathematical disciplines. With over 70,000 electronic transactions per day, the archives are an example of the significance of the electronic means of communication.

13. The E-Print Archives contain many examples of computer code archived with submitted articles. The inclusion of the computer code allows others to verify the results claimed by the scientist publishing the article.

14. For example, one of the archives at E-Print is the astrophysics archive (astro-ph). Archive entry astro-ph/9601144 is The NLTE formation of neutral-boron lines in cool stars. Written by Dan Kiselman, and Mats Carlsson, this article is a theoretical analysis of neutral boron lines in cool stars.

15. To allow others to verify their finding, Mr. Kiselman and Mr. Carlsson include the source codes of the following two computer programs in the archive. The first computer program is named `ccorb_nite.f`, reprinted in Exhibit A. The second computer program is named `corrb_nlte.pro`, reprinted in Exhibit B.

16. Note that these programs are extensively commented to allow other users to understand them and to modify them for additional scientific purposes. Their intrinsic importance to establishing the validity of the authors' scientific results is further evidenced by the authors having made these programs available as well as directly via anonymous FTP (file transfer protocol) from a site in Norway, <ftp://ftp.astro.uio.no/pub/boron>. The above (received during January 1996) is just one example of many received going back to the archive's inception in 1991.

17. The importance of the distribution of software methods and techniques is represented in addition by archive entries that consist entirely of general purpose software and its documentation. For example, archive entry `gr-qc/9502035` (submitted by Harald H. Soleng in February 1995) consists entirely of the computer code and documentation for a program entitled CARTAN. This is a program written in the Mathematica computer language for tensor computations primarily intended for practitioners of the General Theory of Relativity to reproduce and verify one another's calculations.

I declare under penalty of perjury that the foregoing is true and correct.

Dated: \_\_\_\_\_

Dr. Paul Ginsparg **Appendix A - Program `corrb_nlte.f`**

program main

c

c example of a program calling the `corr_nlte` function

c

real `corr_nlte,a12_nlte`

real `a12_lte0,teff0,lgg0,feh0`

real `teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte`

```

integer iline
c
write(*,*) 'line 0: 249.7 nm (air) resonance line'
write(*,*) 'line 1: 209.0 nm (air) subordinate line'
write(*,*) 'EOF (^D) to stop'
100 continue
write(*,*) ' give A12(LTE), Teff, lgg, [Fe/H], line'
read(*,*,end=999) a12_lte0,teff0,lgg0,feh0,iline
a12_nlte=corr_nlte(a12_lte0,teff0,lgg0,feh0,iline)
write(*,*) ' A12(NLTE)=' ,a12_nlte
goto 100
c
999 continue
end
c
c*****
c
function corr_nlte(a12_lte0,teff0,lgg0,feh0,iline)
c+
c NAME:
c corr_nlte
c
c PURPOSE:
c Correct LTE abundance determination of boron for non-LTE
c effects
c
c CATEGORY:
c Astronomy
c
c CALLING SEQUENCE:
c a12_nlte = corr_nlte(a12_lte,teff,lgg,feh,iline)
c
c INPUTS:
c range
c a12_lte [-1.5,4.5] LTE abundance on a12(hydrogen)=12 scale
c actual range often smaller
c teff [5000,7500] Effective temperature
c lgg [2.0,4.5] log(base10) gravity (cgs units)
c feh [-3,0] [Fe/H]
c iline [0,1] line, 0: 249.7 nm (air) resonance line
c 1: 209.0 nm (air) resonance line
c
c data files adiffb0_lte.dat.and.adiffb1_lte.dat are read on first call
c all input arguments must be scalars
c
c OUTPUTS:
c non-LTE abundance (=LTE abundance corrected for non-LTE effects)
c if abscissa is outside range of calculated non-LTE corrections
c a warning message is printed.and.the value -9 is returned.
c
c COMMON BLOCKS:
c /cgrid/ teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte

```

```

c
c SIDE EFFECTS:
c None.
c
c RESTRICTIONS:
c Corrections have been calculated based on Uppsala MARCS models.
c Corrections to LTE abundances calculated from other theoretical
c models may have systematic errors.
c
c PROCEDURE:
c Hermite spline interpolation in a12_lte,
c linear interpolation in other variables.
c
c MODIFICATION HISTORY:
c Mats Carlsson, February 1994
c Dan Kiselman, August 1995 (boron)
c
c-

```

```

real corr_nlte
integer indx
real a12_lte0,teff0,lgg0,feh0
integer iline
logical first_call
save first_call
data first_call/.true./
c
integer na12,nteff,nlgg,nfeh
parameter (na12=21,nteff=6,nlgg=4,nfeh=4)
real teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte
common /cgrid/ teff(nteff),lgg(nlgg),feh(nfeh),a12_lte(na12),
* adiff0_lte(na12,nteff,nlgg,nfeh),adiff1_lte(na12,nteff,nlgg,nfeh)
c
real yp0(2,2,2),y0(na12),yfeh(2,2),ylgg(2),yp
real x(na12),f(na12)
integer ia12,iteff,ilgg,ifeh,iup
integer i,j,k,l,n
c
if(iline.ne.0 .and. iline.ne.1) then
write(*,*) 'iline: 0: 249.7 nm resonance line'
write(*,*) ' 1: 209.0 nm subordinate line'
corr_nlte=-9
return
endif

if(first_call) then
call adiff_read
endif
c test for point inside interpolation box

if(lgg0.lt.lgg(1) .or. lgg0.gt.lgg(nlgg)) then
write(*,*) 'lgg outside range'
corr_nlte=-9

```

```
return
endif
```

```
if(feh0.lt.feh(nfeh) .or. feh0.gt.feh(1)) then
write(*,*) 'feh outside range'
corr_nlte=-9
return
endif
```

```
if(teff0.lt.teff(1) .or. teff0.gt.teff(nteff)) then
write(*,*) 'teff outside range'
corr_nlte=-9
return
endif
```

```
if(a12_lte0.lt.a12_lte(1) .or. a12_lte0.gt.a12_lte(na12)) then
write(*,*) 'a12_lte outside range'
corr_nlte=-9
return
endif
```

c interpolate. index for table abscissae:  
c feh0 is between feh(ifeh).and.feh(ifeh+1) etc  
c note that feh is decreasing while other abscissae are increasing

```
ia12=indx(a12_lte,na12,a12_lte0)
iteff=indx(teff,nteff,teff0)
ilgg=indx(lgg,nlgg,lgg0)
ifeh=indx(feh,nfeh,feh0)
```

c fill interpolation cube in 3D

```
do 400 k=0,1
do 300 j=0,1
do 200 i=0,1
if(iline.eq.0) then
do 100 l=1,na12
y0(l)=adiff0_lte(l,iteff+i,ilgg+j,ifeh+k)
100 continue
else
do 110 l=1,na12
y0(l)=adiff1_lte(l,iteff+i,ilgg+j,ifeh+k)
110 continue
endif
n=n+1
do 120 l=1,na12
if(y0(l).gt.-8.) then
n=n+1
x(n)=a12_lte(l)
f(n)=y0(l)
endif
120 continue
if(n.eq.0) then
```

```

write(*,*) 'no data in range'
corr_nlte=-9
return
endif
if(a12_lte0.lt.x(1) .or. a12_lte0.gt.x(n)) then
write(*,*) 'a12_lte outside range'
corr_nlte=-9
return
endif
call intep(1,x,f,n,iup,a12_lte0,yp0(i+1,j+1,k+1))
200 continue
300 continue
400 continue

```

c interpolate in feh, result in yfeh(2,2)

```

do 600 i=1,2
do 500 j=1,2
yfeh(i,j)=yp0(i,j,1)+(feh0-feh(ifeh))/(feh(ifeh+1)-feh(ifeh))*
* (yp0(i,j,2)-yp0(i,j,1))
500 continue
600 continue

```

c interpolate in lgg, result in ylgg(2)

```

do 700 i=1,2
ylgg(i)=yfeh(i,1)+(lgg0-lgg(ilgg))/(lgg(ilgg+1)-lgg(ilgg))*
* (yfeh(i,2)-yfeh(i,1))
700 continue

```

c interpolate in teff

```

yp=ylgg(1)+(teff0-teff(iteff))/(teff(iteff+1)-teff(iteff))*
* (ylgg(2)-ylgg(1))

```

```

corr_nlte=yp+a12_lte0

```

```

return
end

```

c\*\*\*\*\*

```

function indx(x,n,x0)

```

```

c
c find index such that x0 is between x(i) and x(i+1)
c short arrays, no fancy search
c
integer i,n,indx,iup
real x(n),x0,sign
c
iup=sign(1.0,x(2)-x(1))
do 100 i=2,n

```

```
if(iup*x(i).ge.iup*x0) goto 200
100 continue
200 continue
indx=i-1
return
end
```

```
c*****
```

```
subroutine adiff_read
```

```
c+
```

```
c adiff_read
```

```
c
```

```
c read adiff0_lte.dat and adiff1_lte.dat
```

```
c
```

```
c-
```

```
integer na12,nteff,nlgg,nfeh
```

```
parameter (na12=21,nteff=6,nlgg=4,nfeh=4)
```

```
real teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte
```

```
common /cgrid/ teff(nteff),lgg(nlgg),feh(nfeh),a12_lte(na12),
```

```
* adiff0_lte(na12,nteff,nlgg,nfeh),adiff1_lte(na12,nteff,nlgg,nfeh)
```

```
c
```

```
integer i,j,k,l
```

```
do 100 i=1,nteff
```

```
teff(i)=5000.+(i-1)*500.
```

```
100 continue
```

```
lgg(1)=2.0
```

```
lgg(2)=3.5
```

```
lgg(3)=4.0
```

```
lgg(4)=4.5
```

```
feh(1)= 0.0
```

```
feh(2)=-1.0
```

```
feh(3)=-2.0
```

```
feh(4)=-3.0
```

```
do 200 i=1,na12
```

```
a12_lte(i)=-1.5+(i-1)*0.3
```

```
200 continue
```

```
open(11,file='adiffb0_lte.dat',status='old')
```

```
open(12,file='adiffb1_lte.dat',status='old')
```

```
read(11,*) (((adiff0_lte(l,k,j,i),l=1,na12),k=1,nteff),
```

```
* j=1,nlgg),i=1,nfeh)
```

```
read(12,*) (((adiff1_lte(l,k,j,i),l=1,na12),k=1,nteff),
```

```
* j=1,nlgg),i=1,nfeh)
```

```
close(11)
```

```
close(12)
```

```
end
```

```
c*****
```

```

subroutine intep(it,x,f,n,iup,xp,p)
c
c interpolation routine based on hermite polynomials
c ref: publications of the dominion astrophysical observatory, xvi,6,67
c graham hill: intep, an effective interpolation subroutine
c
real lp1,lp2,l1,l2
dimension f(1),x(1)
if(it.eq.1) then
ier=1
io=1
iup=0
if(x(2).lt.x(1)) iup=1
n1=n-1
if((xp.ge.x(n).and.iup.eq.0).or.(xp.le.x(n).and.iup.eq.1)) then
5 p=f(n)
goto 6
else if((xp.le.x(1).and.iup.eq.0).or.
& (xp.ge.x(1).and.iup.eq.1)) then
p=f(1)
6 ier=2
return
endif
endif
8 do 1 i=io,n
if(xp.lt.x(i).and.iup.eq.0) goto 2
if(xp.gt.x(i).and.iup.eq.1) goto 2
1 continue
goto 5
2 i=i-1
if(i.eq.io-1) goto 4
io=i+1
lp1=1./(x(i)-x(i+1))
lp2=1./(x(i+1)-x(i))
if(i.eq.1) fp1=(f(2)-f(1))/(x(2)-x(1))
if(i.eq.1) goto 3
fp1=(f(i+1)-f(i-1))/(x(i+1)-x(i-1))
3 if(i.ge.n1) fp2=(f(n)-f(n-1))/(x(n)-x(n-1))
if(i.ge.n1) goto 4
fp2=(f(i+2)-f(i))/(x(i+2)-x(i))
4 xpi1=xp-x(i+1)
xpi=xp-x(i)
l1=xpi1*lp1
l2=xpi*lp2
p=f(i)*(1.-2.*lp1*xpi)*l1*l1+f(i+1)*(1.-2.*lp2*xpi1)*
& l2*l2+fp2*xpi1*l2*l2+fp1*xpi*l1*l1
return
end

```

## Appendix B - Program corrb\_nlte.pro

```

function corrb_nlte,a12_lte0,teff0,lgg0,feh0,iline
;+
; NAME:

```

```

; corrb_nlte
;
; PURPOSE:
; Correct LTE abundance determination of boron for non-LTE
; effects
;
; CATEGORY:
; Astronomy
;
; CALLING SEQUENCE:
; a12_nlte = corrb_nlte(a12_lte,teff,lgg,feh,iline)
;
; INPUTS:
; range
; a12_lte [-1.5,4.5] LTE abundance on a12(hydrogen)=12 scale
; actual range often smaller
; teff [5000,7500] Effective temperature
; lgg [2.0,4.5] log(base10) gravity (cgs units)
; feh [-3,0] [Fe/H]
; iline [0,1] line, 0: 249.7 nm (air) resonance line
; 1: 209.0 nm (air) resonance line
;
; data files adiffb0_lte.dat and adiffb1_lte.dat are read on first call
; all input arguments must be scalars
;
; OUTPUTS:
; non-LTE abundance (=LTE abundance corrected for non-LTE effects)
; if abscissa is outside range of calculated non-LTE corrections
; a warning message is printed and the value -9 is returned.
;
; COMMON BLOCKS:
; cgrid,teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte
;
; SIDE EFFECTS:
; None.
;
; RESTRICTIONS:
; Corrections have been calculated based on Uppsala OSMARCS models.
; Corrections to LTE abundances calculated from other theoretical
; models may have systematic errors.
;
; PROCEDURE:
; Hermite spline interpolation in a12_lte,
; linear interpolation in other variables.
;
; MODIFICATION HISTORY:
; Mats Carlsson, February 1994. Code for lithium.
; Dan Kiselman, February 1995. Rewritten for boron.
;
;-

```

```

common cgrid,teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte

```

```

if(n_params() lt 5) then begin
print,'a12_nlte=corrb_nlte(a12_lte,teff,lgg,feh,iline)'
return,-9
endif
if(iline ne 0) and (iline ne 1) then begin
print,'iline: 0: 249.7 nm resonance line'
print,' 1: 209.0 nm resonance line'
return,-9
endif

if(n_elements(teff) eq 0) then begin
adiff_read
endif

; test for point inside interpolation box

if(lgg0 lt min(lgg)) or (lgg0 gt max(lgg)) then begin
print,'lgg outside range'
return,-9
endif

if(feh0 lt min(feh)) or (feh0 gt max(feh)) then begin
print,'feh outside range'
return,-9
endif

if(teff0 lt min(teff)) then begin
print,'teff outside range'
return,-9
endif

if(a12_lte0 lt min(a12_lte)) or (a12_lte0 gt max(a12_lte)) then begin
print,'a12_lte outside range'
return,-9
endif

; interpolate. index for table abscissae:
; feh0 is between feh(ifeh) and feh(ifeh+1) etc
; note that feh is decreasing while other abscissae are increasing

nfeh=n_elements(feh)
nteff=n_elements(teff)
nlgg=n_elements(lgg)
na12=n_elements(a12_lte)

if(feh0 eq feh(0)) then ifeh=0 else ifeh=max(where(feh gt feh0))
if(teff0 eq teff(0)) then iteff=0 else iteff=max(where(teff lt teff0))
if(lgg0 eq lgg(0)) then ilgg=0 else ilgg=max(where(lgg lt lgg0))
if(a12_lte0 eq a12_lte(0)) then $
ia12=0 else ia12=max(where(a12_lte lt a12_lte0))

; fill interpolation cube in 3D

```

```

yp0=fltarr(2,2,2) ; interpolation cube in 3D: teff,lgg,feh
for k=0,1 do begin
for j=0,1 do begin
for i=0,1 do begin
if(iline eq 0) then $
y0=adiff0_lte(*,iteff+i,ilgg+j,ifeh+k) $
else $
y0=adiff1_lte(*,iteff+i,ilgg+j,ifeh+k)
iw=where(y0 gt -8.,count)
if(count le 0) then begin
print,'no data in range'
return,-9
endif
if(a12_lte0 lt min(a12_lte(iw))) or (a12_lte0 gt max(a12_lte(iw))) then begin
print,'a12_lte outside range'
return,-9
endif
intep,a12_lte(iw),y0(iw),a12_lte0,yp
yp0(i,j,k)=yp
endifor
endifor
endifor

; interpolate in feh, result in yfeh(2,2)

yfeh=fltarr(2,2)
for i=0,1 do begin
for j=0,1 do begin
yfeh(i,j)=interpol(yp0(i,j,*),feh(ifeh:ifeh+1),feh0)
endifor
endifor

; interpolate in lgg, result in ylgg(2)

ylgg=fltarr(2)
for i=0,1 do begin
ylgg(i)=interpol(yfeh(i,*),lgg(ilgg:ilgg+1),lgg0)
endifor

; interpolate in teff

y0=interpol(ylgg,teff(iteff:iteff+1),teff0)

return,a12_lte0+y0
end

;*****

pro adiff_read
;+
; adiff_read

```

```

;
; read adiff0_lte.dat and adiff1_lte.dat
;
;-
common cgrid,teff,lgg,feh,a12_lte,adiff0_lte,adiff1_lte

teff=[5000., 5500., 6000., 6500., 7000. ,7500.]
lgg=[2.0, 3.5, 4.0, 4.5]
feh=[0.0, -1.0, -2.0, -3.0]
a12_lte=-1.5+findgen(21)*0.3

adiff0_lte=fltarr(21,6,4,4)
adiff1_lte=fltarr(21,6,4,4)

openr,lur0,'adiffb0_lte.dat',/get_lun
openr,lur1,'adiffb1_lte.dat',/get_lun

readf,lur0,adiff0_lte
readf,lur1,adiff1_lte

free_lun,lur0
free_lun,lur1

end

;*****

pro intep,x,y,xpin,ypout ,nowarning=nowarning
;
; ref: publications of the dominion astrophysical observatory, xvi,6,67
; graham hill: intep, an effective interpolation subroutine
;
if (n_params(0) ne 4) then begin
print,'intep,x,y,xp,yp'
return
endif
if(n_elements(nowarning) eq 0) then nowarning=0
n=n_elements(x)
dum=size(xpin)
if(dum(0) eq 0) then begin
np=1
xp=fltarr(1)
yp=fltarr(1)
xp(0)=xpin
endif else begin
np=dum(1)
xp=xpin
yp=fltarr(np)
endelse

ier=1
io=0
iup=0

```

```

if(x(1) lt x(0)) then iup=1
n1=n-2
for it=0,np-1 do begin
if ((xp(it) ge x(n-1)) and (iup eq 0)) or $
((xp(it) le x(n-1)) and (iup eq 1)) then begin
g5: yp(it)=y(n-1)
goto,g6
endif else begin
if ((xp(it) le x(0)) and (iup eq 0)) or $
((xp(it) ge x(0)) and (iup eq 1)) then begin
yp(it)=y(0)
g6: if (xp(it) ne x(0)) and (xp(it) ne x(n-1)) then ier=2
goto,loop
endif
endelse
g8:
for i=io,n-1 do begin
if(xp(it) lt x(i)) and (iup eq 0) then goto,g2
if(xp(it) gt x(i)) and (iup eq 1) then goto,g2
endfor
goto,g5
g2: i=i-1
if(i eq io-1) then goto,g4
io=i+1
lp1=1./(x(i)-x(i+1))
lp2=1./(x(i+1)-x(i))
if(i eq 0) then fp1=(y(1)-y(0))/(x(1)-x(0))
if(i eq 0) then goto,g3
fp1=(y(i+1)-y(i-1))/(x(i+1)-x(i-1))
g3: if(i ge n1) then fp2=(y(n-1)-y(n-2))/(x(n-1)-x(n-2))
if(i ge n1) then goto,g4
fp2=(y(i+2)-y(i))/(x(i+2)-x(i))
g4: xpi1=xp(it)-x(i+1)
xpi=xp(it)-x(i)
l1=xpi1*lp1
l2=xpi*lp2
yp(it)=y(i)*(1.-2.*lp1*xpi)*l1*l1+y(i+1)*(1.-2.*lp2*xpi1)* $
l2*l2+fp2*xpi1*l2*l2+fp1*xpi*l1*l1
goto,loop
loop:
endfor
result:
if(dum(0) eq 0) then begin
ypout=yp(0)
endif else begin
ypout=yp
endelse
if (ier eq 2) and (nowarning eq 0) then $
print, ' warning: xp outside range, yp set to end-point value'
return
end

```