Reinhart Leitinger, Institut für Geophysik, Astrophysik und Meteorologie, Universität Graz, Austria
and
Sandro Radicella, Abdus Salam ICTP, Trieste, Italy

Software documentation

FORTRAN 77 production programs
electron density model `NeQuick_ITUR`
sample drivers `Eldens_ITUR` and `slQu`

Table of contents

	page
NeQuick_ITUR.for: data files needed	2
NeQuick_ITUR.for: subroutines and functions	2
Sample drivers: subroutines and functions	3
Transfer of variable values via subroutines and functions: Modules of NeQuick_ITUR	4
Transfer of variable values via subroutines and functions: Modules of SlQu	6
Program listing of NeQuick_ITUR.for	8
Program listing of driver eldens_ITUR.for	14
Program listing of driver SlQu.for	16
Short description of the NeQuick electron density model	23
Installation Manual	24
Sample results of eldens_ITUR.exe	24
Sample results of SlQu.exe	24
Use of the model software	28
References	28
Content of the NeQuick model package delivered to ITU-R	29

NeQuick_ITUR.for: data files needed

(a) CCIR "map" files in ASCII mode

`ccir11.asc ... ccir22.asc` (12 files, number 11 for January, number 22 for December)

(b) dip latitudes grid

`diplats.asc`

NeQuick_ITUR.for: subroutines and functions

NeQuick model
```
      real*8 function Nequick(h,alat,along,mth,flx,ut)
```
Entry point for general applications
```
      entry eldens(h,alat,along)
```
Entry point for vertical profile generation
```
      entry vert(h)
```
Bottomside F region model
```
      real*8 function NeMdGR(A,hm,BB,h)
```
Topside F region model
```
      real*8 function topq(h,No,hmax,Ho)
```
Prepare parameter for model
```
      subroutine prepmdgr(mth,R12,foF2,foF1,foE,M3000,Dip,hm,BB,A)
```
"Maps" for foE and foF1
```
      subroutine ef1(alat,mth,flx,chi,foE,foF1)
```
ITU-R (CCIR) map for foF2 and M3000(F2)
```
      subroutine cciri(xMODIP,mth,UT,R12,alat,along,foF2,M3000)
```
Auxiliary function used by cciri
```
      real*8 function gamma1(xMODIP,alat,along,hour,iharm,nq,
     +                          k1,m,mm,m3,sfe)
```
Calculate F2 peak height from ionosonde parameters foE, foF2, M3000(F2)
```
      real*8 function peakh(foE,foF2,M3000)
```
Calculate sine and cosine of the declination of the sun
```
      subroutine sdec(mth,UT,sdelta,cdelta)
```
Read dip latitudes data grid
```
      subroutine geomagin(filenam,pdip)
```
Calculate dip latitude from geographic coordinates
```
      real*8 function philam(pdip,alat,along)
```
Auxiliary module: argument restricted exp function
```
      real*8 function fexp(a)
```
Auxiliary module: joining together of functions f1 and f2 with exponentials
```
      real*8 function djoin(f1,f2,alpha,x)
```
Auxiliary module: third order interpolation
```
      real*8 function finter3(z,x)
```

Sample drivers: subroutines and functions

eldens_IUTR.for : only uses NeQuick, eldens, vert

slQu : uses NeQuick and eldens and contains the following modules

Numerical integration, Gauss procedure, special formulation (integrates `eld`)
```
      real*8 function gint (g1,g2,eps,  pp,Re,sa,ca,ssig,csig,along1)
```
Numerical integration, Gauss procedure, special formulation (integrates `vert`)
```
      real*8 function gintv (g1,g2,eps)
```

Input of ray endpoints and calculation of geometric ray properties
```
      subroutine rays(r1,h1,ph1,alng1,r2,h2,ph2,alng2,zeta,
     & pp,Re,sa,ca,sb,cb,ssig,csig,along1)
```
Input of date, time and solar activity
```
      subroutine dat_t_sa(iyr,mth,nday,ut,R12,flx)
```

Modules called / used by `rays:`

Properties of great circle between ray endpoints
```
      subroutine gcirc(alat1,alat2,along1,along2,s1,c1,s2,c2,ssig,
     + csig,psi)
```
Calculates ray perigee properties, zenith angle of ray, etc., from ray endpoints
```
      subroutine naut(r1,r2,ph1,ph2,alng1,alng2,akappa,
     & pp,php,alamp,zeta,cchi)
```
Height and geographic coordinates from ray coordinate s, special formulation
```
      subroutine geogra(s,  pp,Re,s1,c1,ssig,csig,along1,
     &   h,alat,along)
```

Electron density for numerical integration, special formulation
```
      real*8 function eld(s, pp,Re,s1,c1,ssig,csig,along1)
```

“special formulation”: to avoid common blocks

Transfer of variable values via subroutines and functions

Modules of NeQuick_ITUR:

`real*8 function NeQuick`

(h,	alat,	along,	mth	flx,	UT)
height	latitude	longitude	month	average solar radio flux	Universal Time
km	degrees N	degrees E	1 ... 12	flux units	hours
—— all input ——					
real*8	real*8	real*8	integer	real*8	real*8

`entry eldens`

(h,	alat,	along)
height	latitude	longitude
km	degrees N	degrees E
— all input, all real*8 —		

`entry vert`

(h)
height
km
real*8
input

Subroutines and functions called or used by NeQuick:

`real*8 function NeMdGR`

(A,	hm,	BB,	h)
amplitudes	peak heights	thickness parameters	height
—— Epstein layer parameters ——			
$10^{11}m^{-3}$	km	km	km
—— all input ——			
3*real*8	3*real*8	6*real*8	real*8

`real*8 function topq`

(h,	No,	hmax,	Ho)
height	Peak electron density	peak height	thickness parameter
km	m^{-3}	km	km
—— all input, all real*8 ——			

`subroutine prepMdGR`

(mth,	R12,	foF2,	foF1,	foE,	M3000,	Dip,	hm,	BB	A)
month	sol. activ.	foF2	foF1	foE	M(3000)F2	geomagn. inclination	Epstein layer parameters		
1...12	≥0	MHz	MHz	MHz		degrees	km	km	$10^{11}m^{-3}$
—— input ——							— output —		
integer	—— all real*8 ——						3*r*8	6*r*8	3*r*8

`subroutine ef1`

(alat,	mth,	flx,	chi,	foE,	foF1)
latitude	month	solar radio flux	zenith angle of the sun	foE	foF1
degrees N	1...12	flux units	degrees	MHz	MHz
input	Input	input	input	output	output
real*8	integer	real*8	real*8	real*8	real*8

subroutine cciri

(xMODIP	mth,	ut,	R12,	alat,	along,	foF2,	M3000)
MODIP	Month	Universal Time	solar activity p.	latitude	longitude	foF2	M(3000)F2
degrees	1...12	hours		deg. N	deg, E	MHz	
— — input — —						output	
real*8	integer	— — real*8 — —					

real*8 function gamma1

(xMODIP,	alat,	along,	hour,	iharm,	nq,	k1,m,m,	m3,	sfe)
MODIP	latitude	longitude	UT	order of expansion	const-ants	auxiliary numbers	size of sfe	coeff.
degrees	deg. N	deg. E	hours					
— — all input — —								
— — real*8 — —				integer	k1*i	integer		m3*r*8

subroutine peakh

(foE,	foF2,	M3000)
foE	foF2	M(3000)F2
MHz	MHz	
-- all input, all real*8 --		

subroutine sdec

(ut,	doy,	sdelta,	cdelta)
Universal Time	day of year	sine of declination of sun	cosine of declination of sun
hours			
Input	input	output	output
— — all real*8 — —			

subroutine geomagin

(filenam,	pdip)
	grid point values of dip latitude
	degrees
input	output
character*80	array (0:38,-1:37) of real*8

real*8 function philam

(pdip,	alat,	along)
grid point values of dip latitude	latitude	longitude
degrees	degrees N	degrees E
— — all input — —		
array (0:38,-1:37) of real*8	real*8	real*8

real*8 function fexp

(a)
argument
input
real*8

real*8 function djoin

(f1,	f2,	alpha,	x)
value for x > 0	value for x < 0	steepness parameter	argument
— — all input, all real*8 — —			

real*8 function finter3

(z,	x)
anchor points	argument
— input —	
4*real*8	real*8

Modules of SlQu

`real*8 function gint`

(g1,	g2,	eps,	pp,	Re,	sa,	ca,	ssig,	csig,	along1)
lower limit	upper limit	relative accuracy	ray perigee radius	Earth radius	sine of latitude E1	cosine of latitude E1	sine of azimuth	cosine of azimuth	longitude E1
km	km		km	km					deg. E
— — all input, all real*8 — —									

E1: (lower) ray endpoint 1

`real*8 function gintv`

(g1,	g2,	eps)
lower limit	upper limit	relative accuracy
km	km	
all input, all real*8		

`subroutine rays`

(r1,	h1,	ph1,	alng1,	r2,	h2,	ph2,	alng2,	zeta,
radius E1	height E1	latitude E1	longitude E1	radius E2	height E2	latitude E2	longitude E2	zenith angle E2 from E1
km	km	deg N	deg E	km	km	deg N	deg E	deg
— — all output, all real*8 — —								

E1: (lower) ray endpoint 1, E2: (upper) ray endpoint 2

`subroutine rays, cont.`

pp,	Re,	sa,	ca,	sb,	cb,	ssig,	csig,	along1)
ray perigee radius	Earth radius	sine of latitude E1	cosine of latitude E1	sine of latitude E1	cosine of latitude E1	sine of azimuth	cosine of azimuth	longitude E1
km	km							deg. E
— — all output, all real*8 — —								

E1: (lower) ray endpoint 1, E2: (upper) ray endpoint 2

`subroutine dat_t_sa(iyr,mth,nday,ut,R12,flx)`

(iyr,	mth,	nday,	ut,	R12,	flx)
year	month	day of month	Universal Time	average sunspot number	average solar radio flux
	1 ... 12		hours		flux units
— integer —			— real*8 —		
— — all output — —					

`subroutine gcirc`

(alat1,	alat2,	along1,	along2,
latitude E1	latitude E2	longitude E2	longitude E2
deg N	deg N	deg E	deg E
— all input, all real*8 —			

E1: (lower) ray endpoint 1, E2: (upper) ray endpoint 2

`subroutine gcirc, cont.`

s1,	c1,	s2,	c2,	ssig,	csig,	psi)
sine of latitude E1	cosine of latitude E1	sine of latitude E2	cosine of latitude E2	sine of azimuth	cosine of azimuth	angular distance E2-E1
						deg
— — all output, all real*8 — —						

E1: (lower) ray endpoint 1, E2: (upper) ray endpoint 2 `subroutine naut`

(r1,	r2,	ph1,	ph2,	alng1,	alng2,	akappa,
radius E1	radius E2	latitude E1	latitude E2	longitude E1	longitude E2	ratio `r1/(r1+hi)`
km	km	deg N	deg N	deg E	deg E	
— —			all input, all `real*8`		— —	

E1: (lower) ray endpoint 1, E2: (upper) ray endpoint 2, `hi`: mean ionospheric height (pierce point height)

`subroutine naut, cont.`

pp,	php,	alamp,	zeta,	cchi)
ray perigee radius	ray perigee latitude	ray perigee longitude	zenith angle at E1	cosine of zenith angle at `h=hi` (slant to vertical projection factor)
km	deg N	deg E	deg	
— —		all output, all `real*8`		— —

E1: (lower) ray endpoint 1, E2: (upper) ray endpoint 2

`subroutine geogra`

(s,	pp,	Re,	s1,	c1,	ssig,	csig,	along1,
ray coordinate	ray perigee radius	Earth radius	sine of latitude E1	cosine of latitude E1	sine of azimuth	cosine of azimuth	longitude E1
km	km	km					deg. E
— —			all input, all `real*8`			— —	

E1: (lower) ray endpoint 1

`subroutine geogra, cont.`

h,	alat,	along)
height	latitude	longitude
of raypoint with coordinate s		
km	deg. N	deg E
all output, all `real*8`		

E1: (lower) ray endpoint 1

`real*8 function eld`

(s,	pp,	Re,	s1,	c1,	ssig,	csig,	along1)
ray coordinate	ray perigee radius	Earth radius	sine of latitude E1	cosine of latitude E1	sine of azimuth	cosine of azimuth	longitude E1
km	km	km					deg. E
— —			all output, all `real*8`			— —	

E1: (lower) ray endpoint 1

Program listing of NeQuick_ITUR.for

```
      real*8 function NeQuick(h,alat,along,mth,flx,UT)
      implicit real*8 (a-h,o-z)
      character*11 filenam
      real*8 Nmax,NeMdgr,M3000
      dimension hm(3),A(3),BB(6)
      dimension pdip(0:38,-1:37)

      save

      parameter (pi=3.141592653589793D0)
      parameter (DR=1.74532925199433D-2,RD=5.729577951308232D1)
      if (flx .gt. 193.0D0) flx=193.0D0
      data UT0,mth0,flx0,jdip/-100.0D0,-1,0D0,0/
      if (UT.lt. 0.0D0) UT=UT+24.0D0
      if (UT.ge.24.0D0) UT=UT-24.0D0
      along=dmod(along+360.0D0,360.0D0)
      if (jdip.eq.0) then
         filenam='diplats.asc'
         call geomagin(filenam,pdip)
         jdip=1
      endif
      if (UT.ne.UT0.or.mth.ne.mth0) then
         call sdec(mth,UT,sdelta,cdelta)
         UT0=UT
         mth0=mth
      endif
      if (flx.ne.flx0) then
         R12=sqrt(167273.0D0+(flx-63.7)*1123.6D0)-408.99D0
         flx0=flx
      endif

      entry eldens(h,alat,along)
      xlt=UT0+along/15.0D0
      if (xlt.lt.0.0D0)   xlt=xlt+24.0D0
      if (xlt.ge.24.0D0)  xlt=xlt-24.0D0
      dipl=philam(pdip,alat,along)
      Dip=atan2(2.0*sin(dipl*DR),cos(dipl*DR))
      xMODIP=atan2(Dip,sqrt(abs(cos(alat*DR))))*RD
      Dip=Dip*RD
      cchi=sin(alat*DR)*sdelta+cos(alat*DR)*cdelta*
     * cos(pi*(12.0D0-xlt)/12.0D0)
      chi=atan2(sqrt(1.0D0-cchi*cchi),cchi)*RD
      call cciri(xMODIP,mth0,UT0,R12,alat,along,foF2,M3000)
      call ef1(alat,mth0,flx0,chi,foE,foF1)
      call prepmdgr(mth0,R12,foF2,foF1,foE,M3000,Dip,hm,BB,A)
      Nmax=NeMdgr(A,hm,BB,hm(1))
      entry vert(h)
      if (h.gt.hm(1)) then
         NeQuick=topq(h,Nmax,hm(1),BB(6))
         return
      endif
      NeQuick=NeMdgr(A,hm,BB,h)
      return
      end

      real*8 function NeMdGR(A,hm,BB,h)
      implicit real*8 (a-h,o-z)
      dimension A(3),hm(3),BB(6),B(3)

      save

      parameter (f1=10.0D0,f2=2.0D0)
      parameter (h0=100.0D0)
      parameter (Hd=10.0D0)
      data aN0 /-1.0D0/
```

```
B(1)=BB(5)
B(2)=BB(3)
B(3)=BB(1)
if (h.gt.hm(3)) B(3)=BB(2)
if (h.gt.hm(2)) B(2)=BB(4)
if (h.lt.h0) then
   if (aN0.le.0.0D0) then
      sum=0.0D0
      dsum=0.0D0
      do jj=1,3
         arg0=(h0-hm(jj))
         arg=arg0/B(jj)
         if (jj.gt.1) then
            d=abs(h0-hm(1))
            arg=arg*exp(f1/(1.0D0+f2*d))
         endif
         if (abs(arg).gt.25.0D0) then
            s0=0.0D0
            ds=0.0D0
         else
            ee=exp(arg)
            s0=A(jj)*ee/(1.0D0+ee)**2
            ds=(1.0D0-ee)/(1.0D0+ee)/B(jj)
         endif
         sum=sum+s0
         dsum=dsum+s0*ds
      enddo
      bf=1.0D0-dsum/sum*Hd
      aN0=sum*1.0D11
   endif
   z=(h-h0)/Hd
   NeMdGR=aN0*fexp(1.D0-bf*z-fexp(-z))
   return
else
   sum=0.0D0
   do jj=1,3
      arg0=(h-hm(jj))
      arg=arg0/B(jj)
      if (jj.gt.1) then
         d=abs(h-hm(1))
         arg=arg*exp(f1/(1.0D0+f2*d))
      endif
      if (abs(arg).gt.25.0D0) then
         s0=0.0D0
      else
         ee=exp(arg)
         s0=A(jj)*ee/(1.0D0+ee)**2
      endif
      sum=sum+s0
   enddo
   NeMdGR=sum*1.0D11
   return
endif
end
```

```
real*8 function topq(h,No,hmax,Ho)
implicit real*8 (a-h,o-z)
real*8 No
parameter (g=0.125D0,rfac=100.0D0)
   dh=h-hmax
   g1=g*dh
   z=dh/(Ho*(1.0D0+rfac*g1/(rfac*Ho+g1)))
   ee=fexp(z)
   if (ee.gt.1.0D11) then
      ep=4.0D0/ee
   else
      ep=4.0D0*ee/(1.0D0+ee)**2
   endif
```

```
      topq=No*ep
   return
   end

   subroutine prepmdgr(mth,R12,foF2,foF1,foE,M3000,Dip,hm,BB,A)
   implicit real*8 (a-h,o-z)
   real*8 NmE,NmF1,NmF2,M3000
   dimension A(3),hm(3),BB(6)
   data hmE,B2bot,B1,B1top /120.0D0,40.0D0,40.0D0,40.0D0/

   FNe(X)=0.124*X*X
   FEpst(X,Y,Z,W)=X*fexp((W-Y)/Z)/(1.+fexp((W-Y)/Z))**2

   NmF2=FNe(foF2)
   NmF1=FNe(foF1)
   if(foF1.le.0.0D0.and.foE.gt.2.0D0) NmF1=FNe(foE+0.5D0)
   NmE=FNe(foE)
   hmF1=djoin(108.8D0+14.0D0*NmF1+0.71D0*Dip,
  , 108.8D0+14.0D0*NmF1-0.71D0*Dip,12.0D0,Dip)
   hmF2=peakh(foE,foF2,M3000)
   hm(1)=hmF2
   hm(2)=hmF1
   hm(3)=hmE
   dNdHmx=-3.467D0+0.857D0*log(foF2*foF2)+2.02D0*log(M3000)
   dNdHmx=exp(dNdHmx)*0.01D0
   B2bot=0.385*NmF2/dNdHmx
   A(1)=4.0D0*NmF2
   A(2)=4.0D0*(NmF1-FEpst(A(1),hmF2,B2bot,hmF1))
   A(2)=djoin(A(2),0.05D0,60.0D0,A(2)-0.005D0)
   if(NmF1.le.0.001D0) then
      ax=0.0D0
   else
      ax=A(2)/(0.1D0*NmF1)
   endif
   ax=djoin(ax,1.5D0,20.0D0,ax-1.5D0)
   B1=(hmF2-hmF1)/log(ax)
   B1top=djoin(B2bot+50.0D0,B1,20.0D0,B1-B2bot-50.0D0)
   B1bot=0.7*B1top
   A(3)=4.0D0*(NmE-FEpst(A(2),hmF1,B1bot,hmE)-
  - FEpst(A(1),hmF2,B2bot,hmE))
   A(3)=djoin(A(3),0.005D0,60.0D0,A(3)-0.005D0)
   Betop=0.5D0*B1top
   Bebot=5.0D0
   if (Betop.lt.7.0D0) Betop=7.0D0
   if (mth.gt.3.and.mth.lt.10) then
      b2k=6.705D0-0.014D0*R12-0.008D0*hmF2
   else
      b2k=-7.77+0.097*(hmF2/B2bot)**2+0.153*NmF2
   endif
   b2k=djoin(b2k,2.0D0,1.0D0,b2k-2.0D0)
   b2k=djoin(8.0D0,b2k,1.0D0,b2k-8.0D0)
   B2top=b2k*B2bot
   x=(B2top-150.0D0)/100.0D0
   v=(0.041163D0*x-0.183981D0)*x+1.424472D0
   BB(1)=Bebot
   BB(2)=Betop
   BB(3)=B1bot
   BB(4)=B1top
   BB(5)=B2bot
   BB(6)=B2top
   return
   end

   subroutine ef1(alat,mth,flx,chi,foE,foF1)
   implicit real*8 (a-h,o-z)
   parameter (DR=1.74532925199433D-2)
   parameter (chi0=86.23292796211615D0)

   goto(10,10,20,20,30,30,30,30,20,20,10,10) mth
```

```
10 seas=-1.0D0
   goto 40
20 seas=0.0
   goto 40
30 seas=1.0D0
40 ee=fexp(0.3D0*alat)
   seas=seas*(ee-1.0D0)/(ee+1.0D0)
   chin=djoin(90.0D0-0.24D0*fexp(20.0D0-0.20D0*chi),chi,12.0D0,
  , chi-chi0)
   sfac=(1.112D0-0.019D0*seas)*sqrt(sqrt(flx))
   fa=sfac*fexp(log(cos(chin*DR))*0.3D0)
   foE=sqrt(fa*fa+0.49D0)
   foF1=1.4D0*foE
   foF1=djoin(foF1,0.0D0,12.0D0,chi0-chi)
   return
   end

   subroutine cciri(xMODIP,mth,UT,R12,alat,along,foF2,M3000)
   implicit real*8 (a-h,o-z)
   real*8 M3000
   dimension FF0(988),xm0(441),F2(13,76,2),FM3(9,49,2)
   character*10 filena
   integer QM(7),QF(9)
   save
   data QF/11,11,8,4,1,0,0,0,0/,QM/6,7,5,2,1,0,0/
   data montha,monthb,Rga/13,14,-10.0D0/

   if (mth.ne.montha) then
      write(filena,'(4Hccir,I2.2,4H.asc)') mth+10
      open(77,file=filena,status='OLD',form='FORMATTED')
      read(77,'(4E16.8)') F2,FM3
      close(77)
      montha=mth
   endif
   if (R12.ne.Rga.or.mth.ne.monthb) then
      RR2=R12/100.0D0
      RR1=1.0D0-RR2
      do i=1,76
      do j=1,13
         k=j+13*(i-1)
         FF0(k)=F2(j,i,1)*RR1+F2(j,i,2)*RR2
      enddo
      enddo
      do i=1,49
      do j=1,9
         k=j+9*(i-1)
         xm0(k)=FM3(j,i,1)*RR1+FM3(j,i,2)*RR2
      enddo
      enddo
      Rga=R12
      monthb=mth
   endif
   foF2= gamma1(xMODIP,alat,along,UT,6,QF,9,76,13,988,FF0)
   M3000=gamma1(xMODIP,alat,along,UT,4,QM,7,49, 9,441,xm0)
   return
   end

   real*8 function gamma1(xMODIP,alat,along,hour,iharm,nq,
  ,                          k1,m,mm,m3,sfe)
   implicit real*8 (a-h,o-z)
   real*8 c(12),s(12),coef(100),sum
   dimension nq(k1),xsinx(13),sfe(m3)
   logical numok
   parameter (DR=1.74532925199433D-2)

   hou=(15.0D0*hour-180.0D0)*DR
   s(1)=sin(hou)
   c(1)=cos(hou)
   do i=2,iharm
```

```
   c(i)=c(1)*c(i-1)-s(1)*s(i-1)
   s(i)=c(1)*s(i-1)+s(1)*c(i-1)
enddo
do i=1,m
   mi=(i-1)*mm
   coef(i)=sfe(mi+1)
   do j=1,iharm
      coef(i)=coef(i)+sfe(mi+2*j)*s(j)+sfe(mi+2*j+1)*c(j)
   enddo
enddo
sum=coef(1)
ss=sin(xMODIP*DR)
s3=ss
xsinx(1)=1.0D0
index=nq(1)
do j=1,index
   numok=abs(ss).ge.1.0D-30
   if (numok) then
      sum=sum+coef(1+j)*ss
      xsinx(j+1)=ss
      ss=ss*s3
   else
      xsinx(j+1)=0.0D0
   endif
enddo
if (numok) then
   xsinx(nq(1)+2)=ss
else
   xsinx(nq(1)+2)=0.0D0
endif
np=nq(1)+1
ss=cos(alat*DR)
s3=ss
do j=2,k1
   s0=along*(j-1)*DR
   s1=cos(s0)
   s2=sin(s0)
   index=nq(j)+1
   do L=1,index
      np=np+1
      sum=sum+coef(np)*xsinx(L)*ss*s1
      np=np+1
      sum=sum+coef(np)*xsinx(L)*ss*s2
   enddo
   ss=ss*s3
enddo
gamma1=sum
return
end

real*8 function peakh(foE,foF2,M3000)
implicit real*8 (a-h,o-z)
real*8 MF,M3000
sqM=M3000*M3000
MF=M3000*sqrt((0.0196D0*sqM+1.)/(1.2967D0*sqM-1.0D0))
If(foE.ge.1.0D-30) then
   ratio=foF2/foE
   ratio=djoin(ratio,1.75D0,20.0D0,ratio-1.75D0)
   dM=0.253D0/(ratio-1.215D0)-0.012D0
else
   dM=-0.012D0
endif
peakh=1490.0D0*MF/(M3000+dM)-176.0D0
return
end

subroutine sdec(mth,UT,sdelta,cdelta)
implicit real*8 (a-h,o-z)
parameter (DR=1.74532925199433D-2)
```

```
 doy=mth*30.5D0-15.0D0
 t =doy + (18.0D0-UT)/24.0D0
 amrad=(0.9856D0*t - 3.289D0)*DR
 aLrad = amrad + (1.916D0*sin(amrad)+0.020D0*sin(2.0D0*amrad)+
+ 282.634D0)*DR
 sdelta=0.39782D0*sin(aLrad)
 cdelta=sqrt(1.0D0-sdelta*sdelta)
 return
 end
```

subroutine geomagin(filenam,pdip)

```
 implicit real*8 (a-h,o-z)
 character*11 filenam
 dimension pdip(0:38,-1:37)
 parameter (latp=36,lngp=36,lathp=18,lnghp=18)

 open(77,file=filenam,status='OLD',form='FORMATTED')
 do i=-lnghp,lnghp
    read(77,*) (pdip(i+lnghp+1,j+lathp),j=-lathp,lathp)
 enddo
 close(77)
 do i=0,latp
    pdip(0,i)=pdip(2,mod((i+lathp),latp))
 enddo
 do i=0,latp
    pdip(lngp+2,i)=pdip(lngp-1,mod((i+lathp),latp))
 enddo
 do i=0,lngp+2
    pdip(i,-1)=pdip(i,latp-1)
 enddo
 do i=0,lngp+2
    pdip(i,latp+1)=pdip(i,1)
 enddo
 return
 end
```

real*8 function philam(pdip,alat,along)

```
 implicit real*8 (a-h,o-z)
 dimension pdip(0:38,-1:37)
 dimension z(4),z1(4)
 parameter (lngp=36,dlatp=5.0D0,dlngp=10.0D0)

 dlng1=(along+180.0D0)/dlngp
 dlng1=dlng1-dint(dlng1)
 j1=idint((along+180.0D0)/dlngp)-2
 if (j1.lt.0) j1=j1+lngp
 if (j1.gt.lngp-3) j1=j1-lngp
 a=(alat+90.0D0)/dlatp+1.0D0
 i=idint(a-1.0D-6)-2
 a=a-dfloat(i+2)
 do k = 1,4
    do j=1,4
    z1(j)=pdip(i+j,j1+k)
    enddo
    z(k)=finter3(z1,a)
 enddo
 philam=finter3(z,dlng1)
 return
 end
```

real*8 function fexp(a)

```
 real*8 a
 if(a.gt.80.0D0) then
    fexp=5.5406D34
    return
 endif
 if(a.lt.-80.0) then
    fexp=1.8049D-35
```

```
      return
    endif
    fexp=exp(a)
    return
    end

    real*8 function djoin(f1,f2,alpha,x)
    real*8 f1,f2,alpha,x,ee,fexp
    ee=fexp(alpha*x)
    djoin=(f1*ee+f2)/(ee+1.0D0)
    return
    end

    real*8 function finter3(z,x)
    implicit real*8 (a-h,o-z)
    dimension z(4),a(0:3)

    dx=x*2.0D0-1.0D0
    if (abs(dx+1.0D0).lt.1.0D-10) then
       finter3=z(2)
       return
    else
       g1=(z(3)+z(2))
       g2=(z(3)-z(2))
       g3=(z(4)+z(1))
       g4=(z(4)-z(1))/3.0D0
       a(0)=(9.0D0*g1-g3)
       a(1)=(9.0D0*g2-g4)
       a(2)=(g3-g1)
       a(3)=(g4-g2)
       zi=0.0
       do j=3,0,-1
          zi=zi*dx+a(j)
       enddo
    endif
    finter3=zi/16.0D0
    return
    end
```

eldens_ITUR.for

```
    program elditur
    implicit real*8 (a-h,o-z)
    character*10 filen1
    character*1 fs
    real*8 NeQuick

    filen1='eldens.dat'
    open(16,file=filen1)
    write(6,*)
    write(6,*)'          ****************************************'
    write(6,*)'          *        Test of NeQuick_ITU-R         *'
    write(6,*)'          *   single values and height profile   *'
    write(6,*)'          ****************************************'
    write(6,*)
    write(6,*)'INPUT: month and UT (hours)'
    read(5,*)mth,UT
    write(6,*)
   & 'INPUT: solar activity type:',
   & ' sunspot number (S) or 10.7 cm radio flux (F)?'
    read(5,'(A)')fs
    if (fs.eq.'F'.or.fs.eq.'f') then
       write(6,*)'INPUT: radio flux (>=63 units)'
       read(5,*)flx
       R12=sqrt(167273.0D0+(flx-63.7)*1123.6D0)-408.99D0
    else
       write(6,*)'INPUT: sunspot number (R12)'
       read(5,*)R12
```

```
        flx=63.7D0+(0.728D0+8.9D-4*R12)*R12
      endif
      write(16,'(A,2(F6.1,1H,),I3,1H,,F5.1)')
     + 'S10.7, R12, month, UT: ',flx,R12,mth,ut

      alat=45.0D0
      along=15.0D0
      h=300.0D0
      aNe=NeQuick(h,alat,along,mth,flx,UT)
      write(6,*)
      write(6,*)'NeQuick test 1: electron densities for constant UT'
      write(6,*)
     & 'coordinates loop: end of input: lat > 90 or lat < 90 degrees'
  100 write(6,*)
     & 'INPUT:',
     &' gg. latitude (deg. N), gg. longitude (deg. E), height (km)'
      read(5,*)alat,along,h
      if (abs(alat).gt.90.0D0) goto 110
      aNe=eldens(h,alat,along)
      write(6,'(A,E12.5,A)')
     & ' NeQuick electron density =',aNE,' m^-3'
      goto 100
  110 write(6,*)
      write(6,*)'NeQuick test 2: electron densities for constant LT'
      write(6,*)'INPUT: Local time (LT in hours)'
      read(5,*) xlt
      write(6,*)
  200 write(6,*)
     & 'INPUT:',
     &' gg. latitude (deg. N), gg. longitude (deg. E), height (km)'
      read(5,*)alat,along,h
      if (abs(alat).gt.90.0D0) goto 210
      ut1=xlt-along/15.0d0
      aNe=NeQuick(h,alat,along,mth,flx,ut1)
      write(6,'(A,E12.5,A)')
     & ' NeQuick electron density =',aNE,'  m^-3'
      goto 200
  210 write(6,*)
      write(6,*)'NeQuick test 3: height profile of electron density'
      write(6,*)
     & 'INPUT: gg. latitude (deg. N), gg. longitude (deg. E)'
      read(5,*)alat,along
      write(6,*)
     & 'INPUT:',
     & ' lower height limit, upper height limit, heigt step (all in km)'
      read(5,*)ih1,ih2,idh
      write(16,'(A,F7.2,A,F8.2,A)')
     & '   electron density profile for',alat,'N',along,'E'
      write(16,*)'height electron density'
      write(16,*)' km     m^-3'
      do ih=ih1,ih2,idh
      h=dfloat(ih)
      write(16,'(I5,E12.5)')ih,vert(h)
      enddo

      close(16)
      write(6,*)'Profile output in ',filen1
      end
```

Program listing of driver slQu.for

```
      program slQu
      implicit real*8 (a-h,o-z)
      real*8 NeQuick
      character*80 filen1
      character*1 yn
      parameter (Re=6371.2D0)
      parameter (RD=5.729577951308232D1)

      write(6,*)
      write(6,*)'          ****************************************'
      write(6,*)'          *        Test of NeQuick_ITU-R         *'
      write(6,*)'          *  slant profile and electron content  *'
      write(6,*)'          ****************************************'
      write(6,*)
      write(6,*)
     & 'Electron density is calculated along straight line rays'
      write(6,*)
     & '   from a lower endpoint (1) to a higher one (2).'
      write(6,*)

      filen1='slQu.dat'
      open(16,file=filen1)

      call rays(r1,h1,ph1,alng1,r2,h2,ph2,alng2,zeta,
     & pp,Re,sa,ca,sb,cb,ssig,csig,along1)
      s1=sqrt(r1*r1-pp*pp)
      s2=sqrt(r2*r2-pp*pp)
      write(16,'(A/2F7.2,F9.2)')
     &    'Ray endpoint 1: lat. (deg. N), long. (deg. E), height (km)',
     &     ph1,alng1,h1
      write(16,'(A/2F7.2,F9.2)')
     &    'Ray endpoint 2: lat. (deg. N), long. (deg. E), height (km)',
     &     ph2,alng2,h2
      if (pp.ge.0.1) write(16,'(2A/2F7.2)')
     & 'zenith angle (deg.) and azimuth (N over E to S, deg.)',
     &  ' of ray at endpoint 1 ',zeta,atan2(ssig,csig)*RD

      call dat_t_sa(iyr,mth,nday,ut,R12,flx)

      write(16,'(A,2(F6.1,1H,),I3,1H,,F5.1)')
     + 'S10.7, R12, month, UT: ',flx,R12,mth,ut
      write(16,'(/A/2A)')
     &    'Electron contents along ray.',
     &    '  (h1-h2) means from point in ',
     &    'height h1 to point in height h2 (heights in km)'

      write(6,*)'List electron density profile along ray (y/n)?'
      read(5,'(A)')yn
      h0=0.0D0
      if (h1.gt.h0) h0=h1
      r0=Re+h0
      s0=sqrt(r0*r0-pp*pp)
      aNe=NeQuick(h2,ph1,alng1,mth,flx,UT)
      if (yn.eq.'Y'.or.yn.eq.'y') then
         write(16,*)
     &    's: coordinate along ray, counted from ray perigee'
         write(16,*)
     &    'r: radius (distance from center of Earth)'
         write(16,*)
         write(16,*)'     s       r    height  lat   long   el.density'
         write(16,*)'    km      km      km   deg N  deg E    m^-3'
         dh=10.0D0
         if (h1.ge.500.0D0) dh= 50.0D0
         if (h1.ge.2000.0D0)dh=250.0D0
         h=h1-dh
   10    h=h+dh
```

```
    r=h+Re
    s=sqrt(r*r-pp*pp)
    if (pp.lt.0.1) then
      aNe=vert(h)
    else
      call geogra (s,  pp,Re,sa,ca,ssig,csig,along1,  h,alat,along)
      aNe=eld(s, pp,Re,sa,ca,ssig,csig,along1)
    endif
    write(16,'(3F8.1,2F7.2,E13.6)') s,r,h,alat,along,aNe
    if (nint(h).eq.500)  dh=50.0D0
    if (nint(h).eq.2000) dh=250.0D0
    if (h+dh.le.h2) goto 10
    if (h+0.01D0.lt.h2) then
      r=h2+Re
      s=sqrt(r*r-pp*pp)
      if (pp.lt.0.1) then
        aNe=vert(h2)
      else
        call geogra (s, pp,Re,sa,ca,ssig,csig,along1, h,alat,along)
        aNe=eld(s, pp,Re,sa,ca,ssig,csig,along1)
      endif
      write(16,'(3F8.1,2F7.2,E13.6)') s,r,h2,alat,along,aNe
    endif
 endif

 if (pp.lt.0.1) then
   alat=ph1
   along=alng1
   if (h2.le.1000.0D0) then
     tec1=gintv(h0,h2,1.0D-3)
     write(16,'(A,I4,1H-,I4,1H))')
&      'Electron content (',nint(h0),nint(h2)
     write(16,'(16X,F12.2)')tec1/1.0D12
   else
     h1a=1000.0D0
     if (h2.le.2000.0D0) then
       if (h1.ge.1000.0D0) then
         tec1=gintv(h1,h2,1.0D-3)
         write(16,'(A,I4,1H-,I4,1H) )')
&          'Electron content (',nint(h1),nint(h2)
         write(16,'(16X,F12.2,A)')tec1/1.0D12,'   x10^15 m^-2'
       else
         tec1=gintv(h0, h1a,1.0D-3)
         tec2=gintv(h1a,h2, 1.0D-2)
         tec4=tec1+tec2
         write(16,'(A,2(I4,1H-,I4,3H),(),I4,1H-,I4,1H))')
&          'Electron contents (',nint(h0),nint(h1a),
&                                nint(h1a),nint(h2),
&                                nint(h0), nint(h2)
         write(16,'(16X,3F12.2,A)')
&          tec1/1.0D12,tec2/1.0D12,tec4/1.0D12,'   x10^15 m^-2'
       endif
     else
       if (h1.ge.2000.0D0) then
         tec1=gintv(h1,h2,1.0D-3)
         write(16,'(A,I4,1H-,I5,1H) )')
&          'Electron content (',nint(h1),nint(h2)
         write(16,'(16X,F12.2,A)')tec1/1.0D12,'   x10^15 m^-2'
       else
         h1b=2000.0D0
         if (h1.ge.1000.0D0) then
           tec1=gintv(h1,h1b,1.0D-3)
           tec2=gintv(h1b,h2,1.0D-3)
           tec4=tec1+tec2
           write(16,'(A,I4,1H-,I4,3H),(,I4,1H-,I5,3H),(,
&             I4,1H-,I5,1H))')
&            'Electron contents (',nint(h1),nint(h1b),
&                                  nint(h1b),nint(h2),
&                                  nint(h1), nint(h2)
```

```
            write(16,'(16X,3F12.2,A)')
     &        tec1/1.0D12,tec2/1.0D12,tec4/1.0D12,'   x10^15 m^-2'
          else
            tec1=gintv(h0, h1a,1.0D-3)
            tec2=gintv(h1a,h1b,1.0D-2)
            tec3=gintv(h1b,h2, 1.0D-2)
            tec4=tec1+tec2+tec3
            write(16,'(A,2(I4,1H-,I4,3H),(),I4,1H-,I5,3H),(,
     &                   I4,1H-,I5,1H))')
     &        'Electron contents (',nint(h0),nint(h1a),
     &                            nint(h1a),nint(h1b),
     &                            nint(h1b),nint(h2),
     &                            nint(h0), nint(h2)
            write(16,'(16X,4F12.2,A)')
     &        tec1/1.0D12,tec2/1.0D12,tec3/1.0D12,tec4/1.0D12,
     &        '   x10^15 m^-2'
          endif
        endif
      endif
    endif
  else
    if (h2.le.1000.0D0) then
       tec1=gint(s0,s2,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
       write(16,'(A,I4,1H-,I4,1H))')
     &  'Electron content (',nint(h0),nint(h2)
       write(16,'(16X,F12.2)')tec1/1.0D12
    else
      h1a=1000.0D0
      r1a=h1a+Re
      s1a=sqrt(r1a*r1a-pp*pp)
      if (h2.le.2000.0D0) then
        if (h1.ge.1000.0D0) then
          tec1=gint(s1,s2,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
          write(16,'(A,I4,1H-,I4,1H) )')
     &     'Electron content (',nint(h1),nint(h2)
          write(16,'(16X,F12.2,A)')tec1/1.0D12,'   x10^15 m^-2'
        else
          s2=sqrt(r2*r2-pp*pp)
          tec1=gint(s0, s1a,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
          tec2=gint(s1a,s2, 1.0D-2,  pp,Re,sa,ca,ssig,csig,along1)
          tec4=tec1+tec2
          write(16,'(A,2(I4,1H-,I4,3H),(),I4,1H-,I4,1H))')
     &      'Electron contents (',nint(h0),nint(h1a),
     &                          nint(h1a),nint(h2),
     &                          nint(h0), nint(h2)
          write(16,'(16X,3F12.2,A)')
     &      tec1/1.0D12,tec2/1.0D12,tec4/1.0D12,'   x10^15 m^-2'
        endif
      else
        if (h1.ge.2000.0D0) then
          tec1=gint(s1,s2,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
          write(16,'(A,I4,1H-,I5,1H) )')
     &     'Electron content (',nint(h1),nint(h2)
          write(16,'(16X,F12.2,A)')tec1/1.0D12,'   x10^15 m^-2'
        else
          h1b=2000.0D0
          r1b=h1b+Re
          s1b=sqrt(r1b*r1b-pp*pp)
          if (h1.ge.1000.0D0) then
            tec1=gint(s1,s1b,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
            tec2=gint(s1b,s2,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
            tec4=tec1+tec2
            write(16,'(A,I4,1H-,I4,3H),(,I4,1H-,I5,3H),(,
     &         I4,1H-,I5,1H))')
     &        'Electron contents (',nint(h1),nint(h1b),
     &                            nint(h1b),nint(h2),
     &                            nint(h1), nint(h2)
            write(16,'(16X,3F12.2,A)')
     &        tec1/1.0D12,tec2/1.0D12,tec4/1.0D12,'   x10^15 m^-2'
```

```
          else
            tec1=gint(s0, s1a,1.0D-3,  pp,Re,sa,ca,ssig,csig,along1)
            tec2=gint(s1a,s1b,1.0D-2,  pp,Re,sa,ca,ssig,csig,along1)
            tec3=gint(s1b,s2, 1.0D-2,  pp,Re,sa,ca,ssig,csig,along1)
            tec4=tec1+tec2+tec3
            write(16,'(A,2(I4,1H-,I4,3H),(),I4,1H-,I5,3H),(,
     &                     I4,1H-,I5,1H))')
     &        'Electron contents (',nint(h0),nint(h1a),
     &                              nint(h1a),nint(h1b),
     &                              nint(h1b),nint(h2),
     &                              nint(h0), nint(h2)
            write(16,'(16X,4F12.2,A)')
     &        tec1/1.0D12,tec2/1.0D12,tec3/1.0D12,tec4/1.0D12,
     &        '   x10^15 m^-2'
          endif
        endif
      endif
    endif
  endif

  close(16)
  write(6,*)'Output in ',filen1
  end

  real*8 function gint (g1,g2,eps,pp,Re,s1,c1,ssig,csig,along1)
  implicit real*8 (a-h,o-z)
     n = 8
1    h  =  (g2-g1) / dfloat(n)
     hh = 0.5D0*h
     g  = h*0.5773502691896D0
     y  = g1 + (h-g)*0.5D0
     gint2  =  eld(y,pp,Re,s1,c1,ssig,csig,along1)+
 +             eld(y+g,pp,Re,s1,c1,ssig,csig,along1)
     do m = 1,n-1
        gint2 = gint2 + eld(y+h-g,pp,Re,s1,c1,ssig,csig,along1)+
 +                      eld(y+h,pp,Re,s1,c1,ssig,csig,along1)
        y = y + h
     enddo
     gint2 = gint2*hh
     if (n.eq.8.or.abs(gint1-gint2).gt.eps*abs(gint1)) then
        n = n*2
        gint1 = gint2
        if (n.lt.1024) goto 1
     endif
     gint = gint2+(gint2-gint1)/15.0D0
  return
  end

  real*8 function gintv (g1,g2,eps)
  implicit real*8 (a-h,o-z)
     n = 8
1    h  =  (g2-g1) / dfloat(n)
     hh = 0.5D0*h
     g  = h*0.5773502691896D0
     y  = g1 + (h-g)*0.5D0
     gint2  =  vert(y)+
 +             vert(y+g)
     do m = 1,n-1
        gint2 = gint2 + vert(y+h-g)+
 +                      vert(y+h)
        y = y + h
     enddo
     gint2 = gint2*hh
     if (n.eq.8.or.abs(gint1-gint2).gt.eps*abs(gint1)) then
        n = n*2
        gint1 = gint2
        if (n.lt.1024) goto 1
     endif
     gintv = gint2+(gint2-gint1)/15.0D0
```

```
      return
      end

      subroutine rays(r1,h1,ph1,alng1,r2,h2,ph2,alng2,zeta,
     & pp,Re,s1,c1,s2,c2,ssig,csig,along1)
      implicit real*8 (a-h,o-z)
      akappa=Re/(Re+400.0D0)
   10 write(6,*)'INPUT: ',
     &'Ray endpoint 1: latitude (deg N), longitude (deg E), height (km)'
      read(5,*)ph1,alng1,h1
      write(6,*)'INPUT: ',
     &'Ray endpoint 2: latitude (deg N), longitude (deg E), height (km)'
      read(5,*)ph2,alng2,h2
      if (abs(ph2-ph1).lt.1.0D-5.and.abs(alng2-alng1).lt.1.0D-5) then
         ph2=ph2
         alng2=alng1
      endif
      r1=Re+h1
      r2=Re+h2
      call naut(r1,r2,ph1,ph2,alng1,alng2,akappa,
     &  pp,php,alamp,zeta,cchi)
      if (abs(zeta).gt.90.0.and.pp.lt.Re) then
         write(6,*) ' ray cuts surface of Earth'
         write(6,*) '      or endpoint 2 lower than endpoint 1.'
         write(6,*) ' Repeat input'
         goto 10
      endif
      if (pp.ge.0.1D0)
     & call gcirc(php,ph2,alamp,alng2,s1,c1,s2,c2,ssig,csig,psi)
      along1=alamp
      return
      end

      subroutine dat_t_sa(iyr,mth,nday,ut,R12,flx)
      implicit real*8 (a-h,o-z)
      dimension R12y(12)
      character*1 yn,fs
      nday=15
    1  write(6,*)'INPUT: year, month, UT:'
      read(5,*)iyr,mth,UT
      if (iyr.gt.100.and.(iyr.lt.1931.or.iyr.gt.2049)
     &    .or.iyr.lt.0) then
         write(6,*)
     &    'error in year (valid: 1931-2049 or 0-49 for 2000-2049'
         write(6,*)'   or 50-99 for 1950-1999)'
         write(6,*)'Repeat'
         goto 1
      endif
      if (mth.lt.1.or.mth.gt.12.or.UT.lt.0.or.UT.gt.24) then
         write(6,*)
     &    'input of month or UT not valid (valid: 1-12 and 0-24)'
         write(6,*)'Repeat'
         goto 1
      endif
      if (iyr.lt.50) iyr=iyr+2000
      if (iyr.lt.1900) iyr=iyr+1900
      if (iyr.ge.1931.and.iyr.le.2001) then
         write(6,*)'R12/F10.7 for this year and month (y/n)'
         read(5,'(A)')yn
       else
         yn='N'
       endif
      if (yn.eq.'y'.or.yn.eq.'Y') then
         open(15,file='R12.dat',status='OLD')
   11    read(15,*)j,R12y
         if (j.lt.iyr) goto 11
         close(15)
         R12=R12y(mth)
         flx=63.7D0+(0.728D0+8.9D-4*R12)*R12
```

```
 else
    write(6,*)
&    'INPUT: solar activity type:',
&    ' sunspot number (S) or 10.7 cm radio flux (F)?'
    read(5,'(A)')fs
    if (fs.eq.'F'.or.fs.eq.'f') then
       write(6,*)'INPUT: radio flux (>=63 units)'
       read(5,*)flx
       R12=sqrt(167273.0D0+(flx-63.7)*1123.6D0)-408.99D0
    else
       write(6,*)'INPUT: sunspot number (R12)'
       read(5,*)R12
       flx=63.7D0+(0.728D0+8.9D-4*R12)*R12
    endif
 endif
 if (flx.gt.193.0D0)
&  write(6,'(2A/A/)')
&  ' *** Input solar flux F exceeds 193 units. ',
&  ' Following Recommendation ',
&  '     ITU-R P. 1239, NeQuick limits effective F to 193 units.'
 return
 end
```

subroutine gcirc(alat1,alat2,along1,along2,s1,c1,s2,c2,ssig,

```
& csig,psi)
 implicit real*8 (a-h,o-z)
 parameter (DR=1.74532925199433D-2)
 rlat1=alat1*DR
 rlat2=alat2*DR
 dlong=(along2-along1)*DR
 s1=sin(rlat1)
 s2=sin(rlat2)
 c1=cos(rlat1)
 c2=cos(rlat2)
 sd=sin(dlong)
 cd=cos(dlong)
 if (abs(abs(alat1)-90.0D0).lt.1.0D-10) then
    psi=abs(alat2-alat1)
    ssig=0.0D0
    if (alat1.gt.0.0D0) then
       csig=-1.0D0
    else
       csig=1.0D0
    endif
 else
    cpsi=s1*s2+c1*c2*cd
    spsi=sqrt(1.0D0-cpsi*cpsi)
    ssig=c2*sd/spsi
    csig=(s2-s1*cpsi)/c1/spsi
    psi=atan2(spsi,cpsi)/DR
 endif
 return
 end
```

subroutine naut(r1,r2,ph1,ph2,alng1,alng2,akappa,

```
& pp,php,alamp,zeta,cchi)
 implicit real*8 (a-h,o-z)
 parameter (DR=1.74532925199433D-2,RD=5.729577951308232D1)
 parameter ( pi=3.141592653589793D0)
 if (abs(ph1-ph2).lt.1.0D-5.and.abs(alng1-alng2).lt.1.0D-5) then
    pp=0.0D0
    php=ph1
    alamp=alng1
    zeta=0.0D0
    cchi=1.0D0
 else
    sph1=sin(ph1*DR)
    cph1=cos(ph1*DR)
    sph2=sin(ph2*DR)
```

```
      cph2=cos(ph2*DR)
      cdl12=cos((alng2-alng1)*DR)
      sdl12=sin((alng2-alng1)*DR)
      cdel=sph1*sph2+cph1*cph2*cdl12
      sdel=sqrt(1.0D0-cdel*cdel)
      zeta=atan2(sdel,cdel-r1/r2)
      ssigp=sdl12*cph2/sdel
      csigp=(sph2-cdel*sph1)/sdel/cph1
      delp=-zeta+pi/2.0D0
      sdelp=sin(delp)
      cdelp=cos(delp)
      sphp=sph1*cdelp-cph1*sdelp*csigp
      cphp=sqrt(1.0D0-sphp*sphp)
      php=atan2(sphp,cphp)*RD
      slamp=-ssigp*sdelp/cphp
      clamp=(cdelp-sph1*sphp)/cph1/cphp
      alamp=atan2(slamp,clamp)*RD+alng1
      szeta=sin(zeta)
      pp=r1*szeta
      zeta=zeta*RD
      schi=akappa*szeta
      cchi=sqrt(1.0D0-schi*schi)
   endif
   return
   end
```

subroutine geogra(s, pp,Re,s1,c1,ssig,csig,along1, h,alat,along)

```
   implicit real*8 (a-h,o-z)
   parameter (RD=5.729577951308232D1)
   tdel=s/pp
   cdel=1.0D0/sqrt(1.0D0+tdel*tdel)
   sdel=tdel*cdel
   arg=s1*cdel+c1*sdel*csig
   alat=atan2(arg,sqrt(1.0D0-arg*arg))*RD
   clong=atan2(sdel*ssig*c1,cdel-s1*arg)*RD
   along=clong+along1
   h=sqrt(s*s+pp*pp)-Re
   return
   end
```

real*8 function eld(s,pp,Re,s1,c1,ssig,csig,along1)

```
   implicit real*8 (a-h,o-z)
   parameter (RD=5.729577951308232D1)
   tdel=s/pp
   cdel=1.0D0/sqrt(1.0D0+tdel*tdel)
   sdel=tdel*cdel
   arg=s1*cdel+c1*sdel*csig
   alat=atan2(arg,sqrt(1.0D0-arg*arg))*RD
   clong=atan2(sdel*ssig*c1,cdel-s1*arg)*RD
   along=clong+along1
   h=sqrt(s*s+pp*pp)-Re
   eld=eldens(h,alat,along)
   return
   end
```

Short description of the NeQuick electron density model

NeQuick was developed at ICTP Trieste and at the University of Graz. It is based on the Di Giovanni - Radicella (DGR) model which was modified to the requirements of the COST 238 Action PRIME to give vertical electron content from ground to 1000 km consistent with the COST 238 regional electron content model. The next generation COST model, COSTprof, which was adopted by the COST Action 251 uses a modified DGR in the height range from 100 km to the peak of the F2 layer and a O^+ - H^+ diffusive equilibrium formulation for the topside F layer. The modification of the DGR formulation ensures a true electron density maximum at the F2 peak under all conditions. The topside of NeQuick is a simplified approximation to a diffusive equilibrium, the main improvement over the DGR and COST 238 models being a limited increase with height of the electron density scale height used. NeQuick is a "profiler" which makes use of three profile anchor points: E layer peak (at a fixed height of 120 km), F1 peak, F2 peak. To model the anchor points it uses the "ionosonde parameters" foE, foF1, foF2 (critical frequencies) and M3000(F2) (transfer parameter). For foE we use a model by John Titheridge; foF1 is taken to be proportional to foE during daytime (foF1=1.4*foE) and 0 during nighttime. For foF2 and M3000(F2) we use the ITU-R (CCIR) maps in the mode used by the International Reference Ionosphere (IRI). The bottom side of the electron density profile consists of the superposition of three Epstein layers which peak at the anchor points. The Epstein layers have different thickness parameters for their bottom and top sides (5 "semi-Epstein" layers). The topside of the electron density profile consists of the topside of an Epstein layer with a height dependent thickness parameter. The sub-models contained in NeQuick use monthly average values of solar activity in two forms: average sunspot number R_{12} and average 10.7 cm solar radio flux $F_{10.7}$. The latter is considered to be the primary input parameter. A fixed relation between R_{12} and $F_{10.7}$ is used:

$$\mathbf{F_{10.7} = 63.7 + (0.728+0.00089\ R_{12})\ R_{12}}$$

or

$$\mathbf{R_{12} = [167273 + (\ F_{10.7} - 63.7)1123.6]^{0.5} - 408.99}\ .$$

To conform with Recommendation ITU-R P. 1239 (Point 3: "Prediction of foF2 and M3000(F2)") effective $F_{10.7}$ is limited to 193 units (this corresponds to an R_{12} limit of 150 units).

The inclination of the geomagnetic induction vector (Dip) is also used by NeQuick sub-models. To be consistent with the ITU-R (CCIR) maps the limited spherical harmonics expansion for 1977 was used to calculate a grid point map of dip latitude (input file diplats.asc). NeQuick calculates dip latitude by third order interpolation in geographic latitude and longitude. Dip is calculated from dip latitude.

Installation Manual

Put the following files in the subdirectory for program calling

(1) `ccir11.asc ... ccir22.asc` (12 files)
(2) `diplats.asc`
(3) `R12.dat` (necessary for `slQu` only in case of answer "y" to first question)

Compile **`NeQuick_ITUR.for`** and one of the test drivers
`eldens_ITUR.for` or
`SlQu.for` and link together the object files.

Sample results of `eldens_ITUR.exe`:

For `mth=10, UT=15, R12=150, alat=45, along=15,` lower height limit 100, upper height limit 1000, height step 100 the profile output file `eldens.dat` contains

```
S10.7, R12, month, UT:  192.9, 150.0, 10, 15.0
   electron density profile for  45.00N   15.00E
 height electron density
  km     m^-3
  100  .43525E+11
  200  .29812E+12
  300  .17329E+13
  400  .14825E+13
  500  .74028E+12
  600  .38116E+12
  700  .21876E+12
  800  .13852E+12
  900  .94808E+11
 1000  .68918E+11
```

Sample results of `slQu.exe` (screen prompts: *italics*, input: **bold**)

Ray endpoint 1: latitude (deg N), longitude (deg E), height (km)
45 15 0
Ray endpoint 2: latitude (deg N), longitude (deg E), height (km)
0 35 20000
year, month, UT
1990 10 15
R12/F10.7 for this year and month (y/n)?
y
List electron density profile along ray?
y
Output in slQu.dat

The output file `slQu.dat` contains

```
Ray endpoint 1: lat. (deg. N), long. (deg. E), height (km)
  45.00  15.00      .00
Ray endpoint 2: lat. (deg. N), long. (deg. E), height (km)
    .00  35.00 20000.00
zenith angle (deg.) and azimuth (N over E to S, deg.) of ray at endpoint 1
  60.50 121.58
S10.7, R12, month, UT:  185.1, 142.1, 10, 15.0

Electron contents along ray.
  (h1-h2) means from point in height h1 to point in height h2 (heights in km)
 s: coordinate along ray
 r: radius (distance from center of Earth)

      s       r    height  lat   long   el.density
     km      km      km   deg N  deg E    m^-3
      .0  6371.2      .0  45.00  15.00  .000000E+00
    20.3  6381.2    10.0  44.86  15.10  .000000E+00
    40.4  6391.2    20.0  44.72  15.20  .000000E+00
    60.5  6401.2    30.0  44.58  15.30  .000000E+00
    80.4  6411.2    40.0  44.44  15.40  .000000E+00
   100.3  6421.2    50.0  44.31  15.50  .000000E+00
   120.1  6431.2    60.0  44.17  15.59  .000000E+00
   139.8  6441.2    70.0  44.04  15.69  .168882E+04
   159.4  6451.2    80.0  43.90  15.78  .276334E+09
   178.9  6461.2    90.0  43.77  15.88  .147831E+11
   198.4  6471.2   100.0  43.64  15.97  .426850E+11
   217.7  6481.2   110.0  43.51  16.06  .632287E+11
   237.0  6491.2   120.0  43.37  16.15  .934415E+11
   256.2  6501.2   130.0  43.25  16.23  .106726E+12
   275.3  6511.2   140.0  43.12  16.32  .120959E+12
   294.4  6521.2   150.0  42.99  16.41  .137085E+12
   313.3  6531.2   160.0  42.86  16.49  .156333E+12
   332.2  6541.2   170.0  42.74  16.58  .180142E+12
   351.0  6551.2   180.0  42.61  16.66  .210289E+12
   369.8  6561.2   190.0  42.49  16.74  .249561E+12
   388.4  6571.2   200.0  42.36  16.83  .300536E+12
   407.0  6581.2   210.0  42.24  16.91  .365931E+12
   425.6  6591.2   220.0  42.12  16.99  .448675E+12
   444.0  6601.2   230.0  42.00  17.07  .551621E+12
   462.4  6611.2   240.0  41.88  17.14  .677018E+12
   480.8  6621.2   250.0  41.76  17.22  .825699E+12
   499.0  6631.2   260.0  41.64  17.30  .996004E+12
   517.2  6641.2   270.0  41.52  17.37  .118258E+13
   535.4  6651.2   280.0  41.40  17.45  .137541E+13
   553.5  6661.2   290.0  41.28  17.52  .155962E+13
   571.5  6671.2   300.0  41.17  17.60  .171659E+13
   589.4  6681.2   310.0  41.05  17.67  .182641E+13
   607.3  6691.2   320.0  40.94  17.74  .187501E+13
   625.2  6701.2   330.0  40.82  17.81  .187361E+13
   642.9  6711.2   340.0  40.71  17.89  .184689E+13
   660.7  6721.2   350.0  40.60  17.96  .180074E+13
   678.3  6731.2   360.0  40.48  18.03  .173944E+13
   695.9  6741.2   370.0  40.37  18.09  .166709E+13
   713.5  6751.2   380.0  40.26  18.16  .158738E+13
   731.0  6761.2   390.0  40.15  18.23  .150346E+13
   748.5  6771.2   400.0  40.04  18.30  .141791E+13
   765.9  6781.2   410.0  39.93  18.36  .133276E+13
   783.2  6791.2   420.0  39.82  18.43  .124952E+13
   800.5  6801.2   430.0  39.72  18.49  .116928E+13
   817.7  6811.2   440.0  39.61  18.56  .109275E+13
   834.9  6821.2   450.0  39.50  18.62  .102038E+13
   852.1  6831.2   460.0  39.40  18.69  .952411E+12
   869.2  6841.2   470.0  39.29  18.75  .888890E+12
   886.2  6851.2   480.0  39.19  18.81  .829768E+12
   903.2  6861.2   490.0  39.08  18.87  .774908E+12
   920.2  6871.2   500.0  38.98  18.93  .724118E+12
  1004.3  6921.2   550.0  38.47  19.23  .523001E+12
```

1087.3	6971.2	600.0	37.97	19.52	.388618E+12
1169.3	7021.2	650.0	37.48	19.80	.297679E+12
1250.4	7071.2	700.0	37.01	20.06	.234728E+12
1330.5	7121.2	750.0	36.55	20.32	.190043E+12
1409.7	7171.2	800.0	36.09	20.56	.157547E+12
1488.2	7221.2	850.0	35.65	20.80	.133380E+12
1565.9	7271.2	900.0	35.22	21.03	.115043E+12
1642.8	7321.2	950.0	34.80	21.25	.100878E+12
1719.0	7371.2	1000.0	34.39	21.46	.897655E+11
1794.6	7421.2	1050.0	33.98	21.67	.809308E+11
1869.5	7471.2	1100.0	33.59	21.87	.738262E+11
1943.8	7521.2	1150.0	33.20	22.07	.680576E+11
2017.5	7571.2	1200.0	32.82	22.26	.633359E+11
2090.7	7621.2	1250.0	32.45	22.44	.594462E+11
2163.3	7671.2	1300.0	32.08	22.62	.562258E+11
2235.4	7721.2	1350.0	31.73	22.79	.535497E+11
2307.0	7771.2	1400.0	31.38	22.96	.513210E+11
2378.1	7821.2	1450.0	31.03	23.13	.494628E+11
2448.8	7871.2	1500.0	30.69	23.29	.479139E+11
2519.0	7921.2	1550.0	30.36	23.44	.466246E+11
2588.9	7971.2	1600.0	30.04	23.59	.455538E+11
2658.2	8021.2	1650.0	29.72	23.74	.446654E+11
2727.3	8071.2	1700.0	29.40	23.89	.439324E+11
2795.9	8121.2	1750.0	29.09	24.03	.433301E+11
2864.1	8171.2	1800.0	28.79	24.17	.428371E+11
2932.0	8221.2	1850.0	28.49	24.30	.424342E+11
2999.6	8271.2	1900.0	28.20	24.43	.421045E+11
3066.8	8321.2	1950.0	27.91	24.56	.418323E+11
3133.7	8371.2	2000.0	27.63	24.69	.416036E+11
3463.7	8621.2	2250.0	26.27	25.28	.406907E+11
3787.0	8871.2	2500.0	25.02	25.82	.393019E+11
4104.5	9121.2	2750.0	23.85	26.30	.367127E+11
4417.0	9371.2	3000.0	22.77	26.75	.328152E+11
4724.9	9621.2	3250.0	21.75	27.16	.280045E+11
5029.0	9871.2	3500.0	20.80	27.54	.229344E+11
5329.4	10121.2	3750.0	19.90	27.89	.182208E+11
5626.8	10371.2	4000.0	19.05	28.22	.142508E+11
5921.2	10621.2	4250.0	18.26	28.52	.111468E+11
6213.1	10871.2	4500.0	17.50	28.81	.884227E+10
6502.6	11121.2	4750.0	16.78	29.08	.718261E+10
6789.9	11371.2	5000.0	16.10	29.33	.600074E+10
7075.3	11621.2	5250.0	15.46	29.57	.515462E+10
7358.9	11871.2	5500.0	14.84	29.80	.453779E+10
7640.9	12121.2	5750.0	14.26	30.01	.407617E+10
7921.3	12371.2	6000.0	13.70	30.22	.372041E+10
8200.2	12621.2	6250.0	13.16	30.41	.343782E+10
8477.9	12871.2	6500.0	12.65	30.60	.320719E+10
8754.3	13121.2	6750.0	12.16	30.77	.301438E+10
9029.6	13371.2	7000.0	11.69	30.94	.284966E+10
9303.8	13621.2	7250.0	11.24	31.10	.270622E+10
9577.0	13871.2	7500.0	10.80	31.26	.257909E+10
9849.3	14121.2	7750.0	10.39	31.41	.246467E+10
10120.7	14371.2	8000.0	9.98	31.55	.236022E+10
10391.3	14621.2	8250.0	9.60	31.68	.226373E+10
10661.1	14871.2	8500.0	9.23	31.82	.217366E+10
10930.2	15121.2	8750.0	8.87	31.94	.208888E+10
11198.6	15371.2	9000.0	8.52	32.06	.200852E+10
11466.3	15621.2	9250.0	8.18	32.18	.193195E+10
11733.4	15871.2	9500.0	7.86	32.29	.185868E+10
11999.9	16121.2	9750.0	7.55	32.40	.178837E+10
12265.9	16371.2	10000.0	7.25	32.51	.172076E+10
12531.4	16621.2	10250.0	6.95	32.61	.165566E+10
12796.3	16871.2	10500.0	6.67	32.71	.159291E+10
13060.8	17121.2	10750.0	6.39	32.80	.153241E+10
13324.8	17371.2	11000.0	6.13	32.90	.147408E+10
13588.4	17621.2	11250.0	5.87	32.99	.141782E+10
13851.6	17871.2	11500.0	5.62	33.07	.136360E+10
14114.3	18121.2	11750.0	5.37	33.16	.131134E+10
14376.8	18371.2	12000.0	5.14	33.24	.126100E+10

```
 14638.8 18621.2 12250.0   4.91  33.32  .121253E+10
 14900.5 18871.2 12500.0   4.68  33.39  .116587E+10
 15161.9 19121.2 12750.0   4.47  33.47  .112098E+10
 15423.0 19371.2 13000.0   4.25  33.54  .107780E+10
 15683.7 19621.2 13250.0   4.05  33.61  .103629E+10
 15944.2 19871.2 13500.0   3.85  33.68  .996383E+09
 16204.4 20121.2 13750.0   3.65  33.75  .958039E+09
 16464.4 20371.2 14000.0   3.46  33.82  .921202E+09
 16724.1 20621.2 14250.0   3.27  33.88  .885819E+09
 16983.5 20871.2 14500.0   3.09  33.94  .851840E+09
 17242.7 21121.2 14750.0   2.92  34.00  .819212E+09
 17501.7 21371.2 15000.0   2.74  34.06  .787884E+09
 17760.4 21621.2 15250.0   2.58  34.12  .757807E+09
 18019.0 21871.2 15500.0   2.41  34.17  .728931E+09
 18277.3 22121.2 15750.0   2.25  34.23  .701209E+09
 18535.5 22371.2 16000.0   2.09  34.28  .674593E+09
 18793.4 22621.2 16250.0   1.94  34.34  .649040E+09
 19051.2 22871.2 16500.0   1.79  34.39  .624504E+09
 19308.8 23121.2 16750.0   1.64  34.44  .600944E+09
 19566.2 23371.2 17000.0   1.50  34.49  .578319E+09
 19823.5 23621.2 17250.0   1.36  34.53  .556589E+09
 20080.6 23871.2 17500.0   1.22  34.58  .535717E+09
 20337.6 24121.2 17750.0   1.09  34.63  .515666E+09
 20594.4 24371.2 18000.0    .96  34.67  .496402E+09
 20851.0 24621.2 18250.0    .83  34.72  .477891E+09
 21107.6 24871.2 18500.0    .70  34.76  .460102E+09
 21364.0 25121.2 18750.0    .58  34.80  .443003E+09
 21620.2 25371.2 19000.0    .46  34.84  .426567E+09
 21876.3 25621.2 19250.0    .34  34.88  .410764E+09
 22132.3 25871.2 19500.0    .23  34.92  .395569E+09
 22388.2 26121.2 19750.0    .11  34.96  .380957E+09
 22644.0 26371.2 20000.0    .00  35.00  .366902E+09
Electron contents (   0-1000),(1000-2000),(2000-20000),(   0-20000)
                      882.61       76.31      114.39     1073.31   x10^15 m^-2
```

Use of the model software

The model function
`NeQuick(h,alat,along,mth,flx,ut)`
has to be used once for each (new) set of season (month), time (UT) and solar activity (10.7 cm solar radio flux) data.
If no change in these data occurs it is sufficient to use the entry point
`eldens(h,alat,along)`
for each new set of coordinates (height, gg. latitude, gg. longitude) or the entry point
`vert(h)`
if the geographic latitude and longitude remain constant.

Since the sub-models of NeQuick need both average sunspot number and 10.7 cm solar radio flux NeQuick_ITUR uses internally a fixed relation
$R_{12} = (F_{10.7} - 57)/0.93$
It is recommended to use this relation to convert R_{12} into $F_{10.7}$.
Any numerical integration routine can be used to gain electron content along straight line rays. The routine **gint** in the driver **slQu** is based on a third order Gauss algorithm. It has been prepared to integrate the "common block free" version of the function **eld.** For mass use (integration along many rays) it is recommended to use the flexible formulation of **gint** which takes the function to be integrated as an input parameter. Since in this case the function to be integrated needs to have a "mathematical" formulation with one input parameter only, common blocks cannot be avoided with FORTRAN 77. Adequate software is available on request from
reinhart.leitinger@uni-graz.at

References

Di Giovanni, G., S. R. Radicella, An analytical model of the electron density profile in the ionosphere, Adv. Space Res. **10**, 27-30, 1990
Leitinger, R., J.E. Titheridge, G. Kirchengast, W. Rothleitner, A „simple“ global empirical model for the F layer of the ionosphere. Wiss. Bericht 1/1995, IMG Uni Graz, 1995
Leitinger, R. and G. Kirchengast, Easy to use global and regional ionospheric models - a report on approaches used in Graz, Acta Geodet. Geophys. Hung. **32**, 329-342, 1997
Leitinger, R. S. Radicella, B. Nava, Electron density models for assessment studies - new developments, Acta Geodet. Geophys. Hung. **37**, 183-193, 2002
Radicella, S.M., M.-L. Zhang, The improved DGR analytical model of electron density height profile and total electron content in the ionosphere. A. Geofisica, **38**, 35-41, 1995

Content of the NeQuick model package delivered to ITU-R

1. FORTRAN 77 Source codes
   1.1. NeQuick_ITUR.for
   1.2. eldens_ITUR.for
   1.3. SlQu.for
2. PC DOS executables
   2.1. eldens_ITUR.exe
   2.2. SlQu.ITUR.exe
3. Data files
   3.01. ccir11.asc
   3.02. ccir12.asc
   3.03. ccir13.asc
   3.04. ccir14.asc
   3.05. ccir15.asc
   3.06. ccir16.asc
   3.07. ccir17.asc
   3.08. ccir18.asc
   3.09. ccir19.asc
   3.10. ccir20.asc
   3.11. ccir21.asc
   3.12. ccir22.asc
   3.13. diplats.asc
   3.14. R12.dat
4. This document