

La correzione del puntamento per rifrazione

Tonino Pisanu, Carlo Migoni, Giuseppe Maccaferri.

Rapporto Interno IRA N° 329/02

Revisori: Prof. Gavril Grueff, Ing. Alessandro Orfei

Indice

Introduzione

1	Teoria della rifrazione	2
2	La rifrazione nel Field System	5
3	Confronto con altre routines	8
4	Appendice	15

Sommario

Durante le varie campagne di misura degli errori di puntamento causati da effetti termici, effettuate a Medicina con il ricevitore a 22 GHz in fuoco primario, sono stati presi in considerazione tutti i vari fattori che influenzavano il corretto puntamento dell'antenna.

Fra questi, abbiamo analizzato quale fosse l'influenza della correzione per la rifrazione atmosferica.

Da una analisi della routine utilizzata, era apparso che la correzione introdotta era in discordanza con l'andamento che ci si aspettava e con la correzione ottenuta con altri programmi. In questo report, descriviamo come funziona la correzione e mostriamo quali sono le differenze fra le diverse routine testate.

Capitolo 1

Teoria della rifrazione

L'atmosfera che circonda la terra, provoca la curvatura della radiazione elettromagnetica che la attraversa, determinando uno spostamento della posizione apparente delle sorgenti.

Il fenomeno della rifrazione segue la ben nota legge di Snell:

$$n_1 \cdot \sin(\theta_1) = n_2 \cdot \sin(\theta_2) \quad (1.1)$$

dove n_1 e n_2 sono gli indici di rifrazione dei due mezzi attraversati dalla radiazione e θ_1 e θ_2 gli angoli di incidenza.

La rifrazione produce un errore in elevazione ΔE fra l'elevazione osservata e quella vera, che può essere espressa in forma semplificata come[2]:

$$\Delta E = E_{obs} - E_{true} = C(n - 1)f(E_{obs}) \quad (1.2)$$

la E_{obs} è l'elevazione osservata alla quale si trova la sorgente mentre E_{true} è l'elevazione vera, senza la curvatura introdotta dall'atmosfera. C è una costante che può essere ricavata o dalla teoria oppure da fit ai minimi quadrati sui dati osservati. n è l'indice di rifrazione dell'aria che varia lungo la linea di vista al variare delle condizioni meteorologiche. $f(E_{obs})$ è una funzione ottenuta con modelli che descrivono il comportamento dell'atmosfera al variare dell'elevazione e che in prima approssimazione può essere rappresentata dalla funzione $\cot(E)$.

Il valore di ΔE ad una elevazione di 5° è $\Delta E \sim 600''$, mentre le condizioni meteo possono alterare il valore di $(n_0 - 1)$ e ΔE di $\pm 15\%$

Una delle approssimazioni principali è dunque quella di ricostruire il profilo dell'atmosfera a tutte le altitudini, utilizzando i valori meteo acquisiti a terra.

Nella regione radio, la rifrazione dipende dalla parte reale dell'indice di rifrazione dei gas che compongono l'atmosfera e risulta quasi indipendente dalla lunghezza d'onda.

Per poter ottenere dei numeri facilmente maneggiabili, introduciamo una nuova grandezza, la rifrattività N che risulta essere legata all'indice di rifrazione n dalla formula seguente[1]:

$$N = \text{Re}[n - 1] \times 10^6 \quad (1.3)$$

La rifrattività dipende dalla composizione chimica dell'atmosfera, secondo la relazione[1] seguente:

$$N = 77.6 \frac{P}{T} + 3.73 \times 10^5 \frac{e}{T^2} \quad (1.4)$$

In questa equazione, P è la pressione totale misurata dal barometro, e la pressione parziale del vapor d'acqua e T la temperatura assoluta dell'aria.

La formula (1.4) è una forma approssimata di una formula più generale che contiene anche le pressioni parziali degli altri gas principali che compongono l'atmosfera, quali il CO_2 che però vengono trascurati poichè danno un contributo molto piccolo (solo $\sim 0.03\%$) alla pressione totale.

La pressione parziale di vapor d'acqua può essere ricavata in diversi modi, a seconda del tipo di sensori che si usano nella stazione meteo. Nel nostro caso, abbiamo a disposizione un barometro per la misura della pressione, un termometro e un igrometro per la misura dell'umidità relativa.

La pressione di vapore è legata alla umidità relativa U , dalla formula seguente[1]:

$$e \simeq e_s(T) \frac{U}{100} \quad (1.5)$$

dove $e_s(T)$ è la pressione di vapor saturo in millibar che si può ricavare in forma approssimata dalla formula seguente[1]:

$$e_s(T) = 6.105 \exp\left[25.22\left(\frac{T - 273}{T}\right) - 5.31 \ln\left(\frac{T}{273}\right)\right] \quad (1.6)$$

Esistono altre formule per il calcolo della correzione per rifrazione che danno risultati molto simili a quelli trovati utilizzando queste equazioni in forma semplificata.

Per capire quali debbano essere le caratteristiche dei sensori meteo da usare, mostriamo nella tabella seguente, quali debbano essere le accuratezze del barometro (σ_p), del termometro (σ_{T_c}) e dell'igrometro (σ_{HD}), se si vuole avere una precisione di puntamento di 1 arcsecondo, ad una elevazione di 15° .

El= 15°	$\sigma_p = \pm 2 \text{ mmHg}$ $\sigma_{T_c} = \pm 0.6^\circ \text{ C}$ $\sigma_{HD} = \pm 1.4 \%$
----------------	---

La stazione meteo utilizzata a Medicina, è composta da: un igrometro con sensore a capacità e accuratezza del $\pm 2\%$, un termometro con accuratezza di $\pm 0.2^\circ \text{ C}$ e un barometro con accuratezza di $\pm 1.5 \text{ mmHg}$. Il sensore più critico dei tre è l'igrometro che ogni tanto necessita di essere ricalibrato per ottenere l'accuratezza dichiarata.

Capitolo 2

La rifrazione nel Field System

Il Field System (FS) prevede che vi sia una sezione *st*(Station Programs), in cui ogni singola stazione ripone i programmi necessari al proprio funzionamento e che non sono presenti nel FS perchè necessitano di essere adattati alla strumentazione o configurazione locali.

La routine che veniva utilizzata nel FS si chiama *refrw*, è scritta in C e ha il listato mostrato in appendice.

In questa routine purtroppo non ci sono molti commenti alle formule utilizzate e risulta di difficile interpretazione la parte riguardante il calcolo della pressione parziale di vapor d'acqua per il quale viene utilizzata la variabile *dewpt*. In particolare merita attenzione anche la variabile *cvt* che nel nostro caso non deve essere utilizzata poichè la nostra stazione misura la pressione direttamente in mbar e non in millimetri di mercurio (mmHg).

Per verificare la bontà della routine, l'abbiamo provata utilizzando le formule viste in precedenza prese dal Crane per il calcolo della pressione di vapor saturo e quindi della pressione parziale di vapore e riscritto la parte di programma per il calcolo della pressione parziale come segue:

```
/* Compute SN (surface refractivity) */
```

```
tempk = tempc + 273.16;
```

```
pwpst=6.105*exp(25.22*(tempc/tempk)-5.31*log(tempk/273)); /*pressione di vapor saturo
```

```
pp=pwpst*humi/100; /*Pressione parziale di vapor d'acqua*/
```

```
sn = 77.6*(pres+(4810.0*pp)/tempk)/tempk; /*Rifrattivita'*/
```

Mostriamo di seguito i grafici che fanno vedere le differenze dei risultati della *refrw* con quelli della nuova *refrw* modificata con le formule del Crane[1].

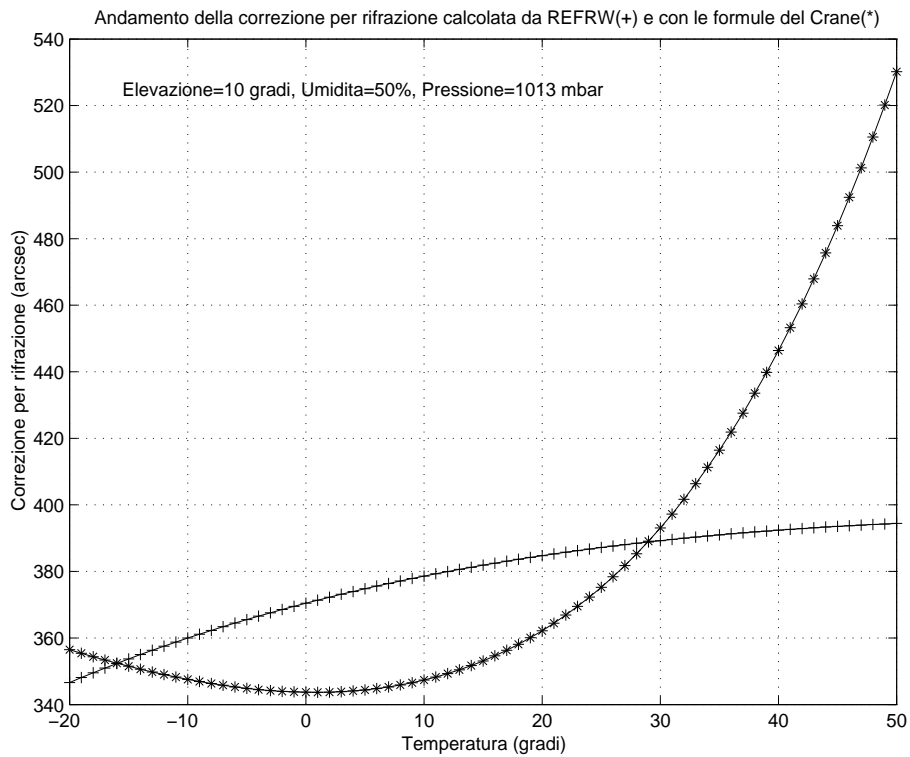


Figura 2.1: Andamento della correzione del Field System e del Crane

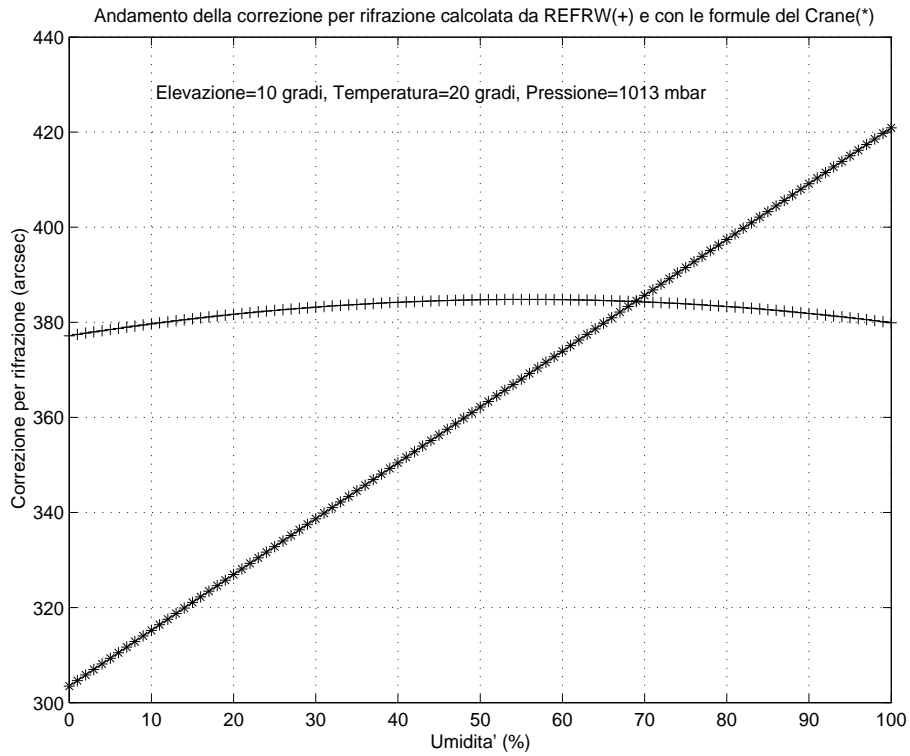


Figura 2.2: Andamento della correzione del Field System e del Crane

Non mostriamo il grafico della correzione in funzione della pressione, poichè le nuove formule che abbiamo introdotto non hanno nessuna dipendenza da questa e quindi l'andamento è rimasto lo stesso.

Capitolo 3

Confronto con altre routines

Una volta viste le differenze fra la nuova e la vecchia routine, abbiamo verificato se la correzione introdotta con le nuove formule, fosse confrontabile con altri algoritmi utilizzati presso altre stazioni.

Per confermare la correttezza del nuovo algoritmo ne abbiamo provato altri due, uno che viene utilizzato nelle antenne del DSN (Deep Space Network) e l'altra denominata *refro* che fa parte della libreria astronomica *SLALIB Positional Astronomy Library*[5]. Entrambe queste routine erano in completo accordo con quella che abbiamo modificato partendo dalla *refrw*, a conferma che quella utilizzata in precedenza era affetta da notevoli errori.

Per quanto riguarda la *refro* abbiamo utilizzato i seguenti valori per le variabili in input di cui ha bisogno la routine:

```
hm=25;      /*Altitudine (metri) sul livello del mare*/
wl=200;     /*Lunghezza d'onda (micrometri)(< 100 per l'ottico)*/
t1r=0.0065; /*Gradiente termico verticale (gradi/metro)*/
eps=1e-8;   /*Precisione richiesta per la iterazione (radianti) */
phi=44.31*DEG2RAD;/*Latitudine del sito (radianti)*/
```

Come abbiamo detto in precedenza, nella regione radio non si ha una sensibile dipendenza con la lunghezza d'onda, per cui per usare il programma nella regione radio è sufficiente usare un valore di *wl* maggiore di 100.

Prima di vedere le dipendenze con i parametri meteo, abbiamo confrontato le correzioni alle stesse condizioni meteo (Temperatura=10 gradi, Umidità=50 %, Pressione=1013 mbar) per vedere se ci fossero delle differenze, i risultati in arcosecondi sono mostrati nella tabella seguente:

Elevazione	Correzione refrw	Corr. Crane	Corr. SLALIB	Corr. DSN
10	384	362	361	357
20	190	179	179	184
30	121	114	113	114
40	83	78	78	78
50	59	55	55	56
60	40	38	38	38
70	25	24	24	23
80	12	12	12	12

Come si può vedere dalla tabella, le differenze fra la correzione introdotta dalla routine refrw e le altre sono di parecchi arcosecondi fino ad elevazioni di 60 gradi.

Vediamo di seguito i plot che mostrano le correzioni introdotte al variare di pressione, temperatura e umidità.

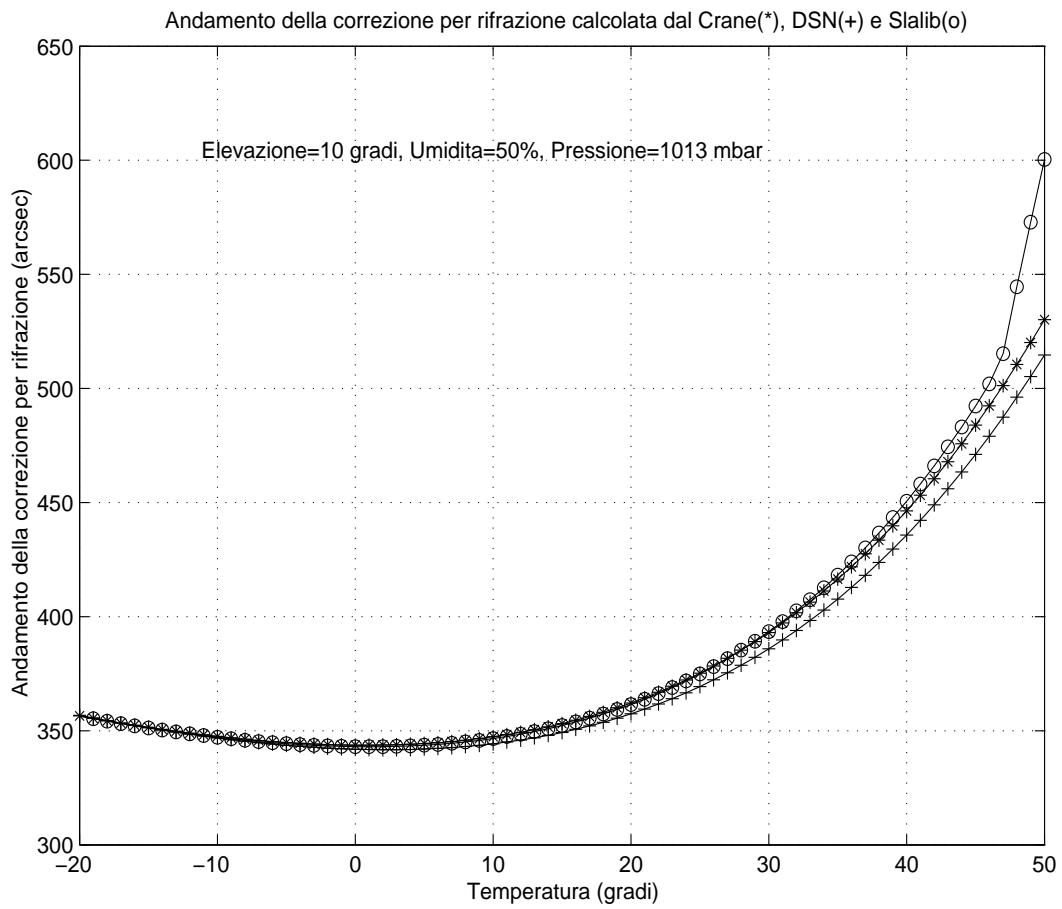


Figura 3.1: Confronto fra la nuova correzione del Crane, del DSN e della SLALIB

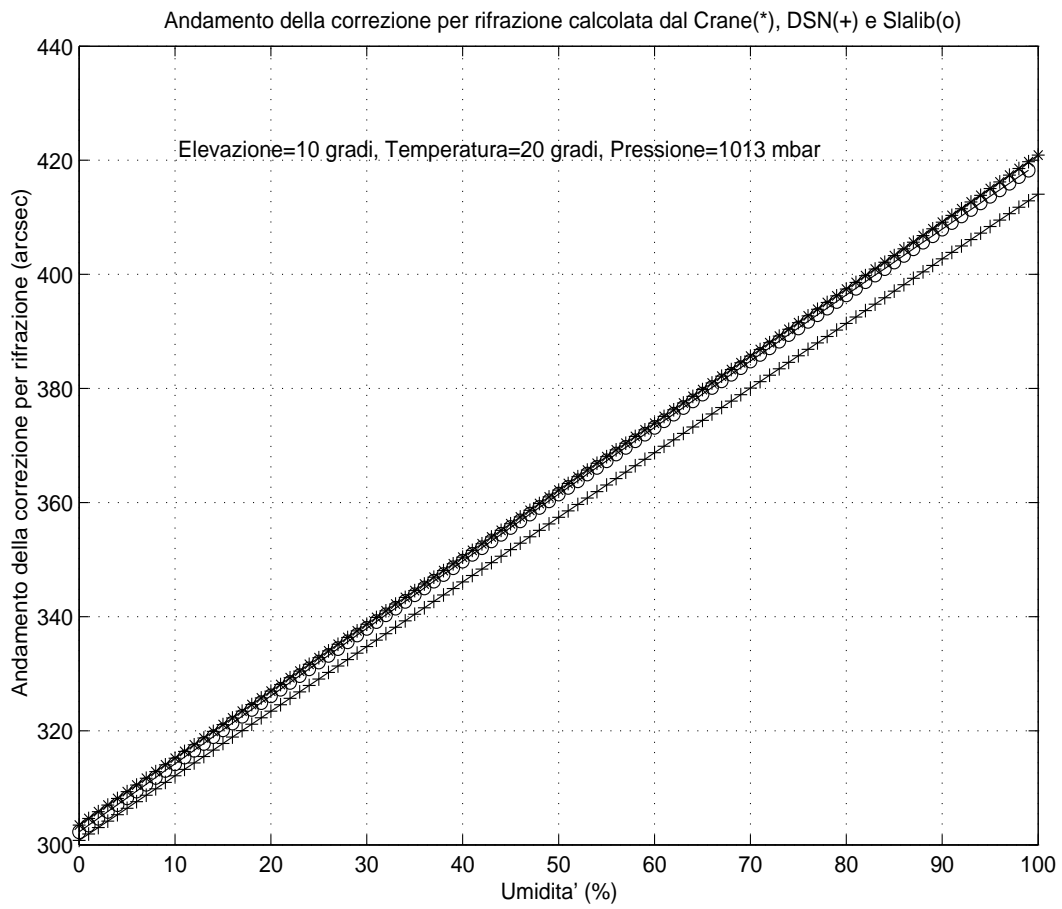


Figura 3.2: Confronto fra la nuova correzione del Crane, del DSN e della SLALIB

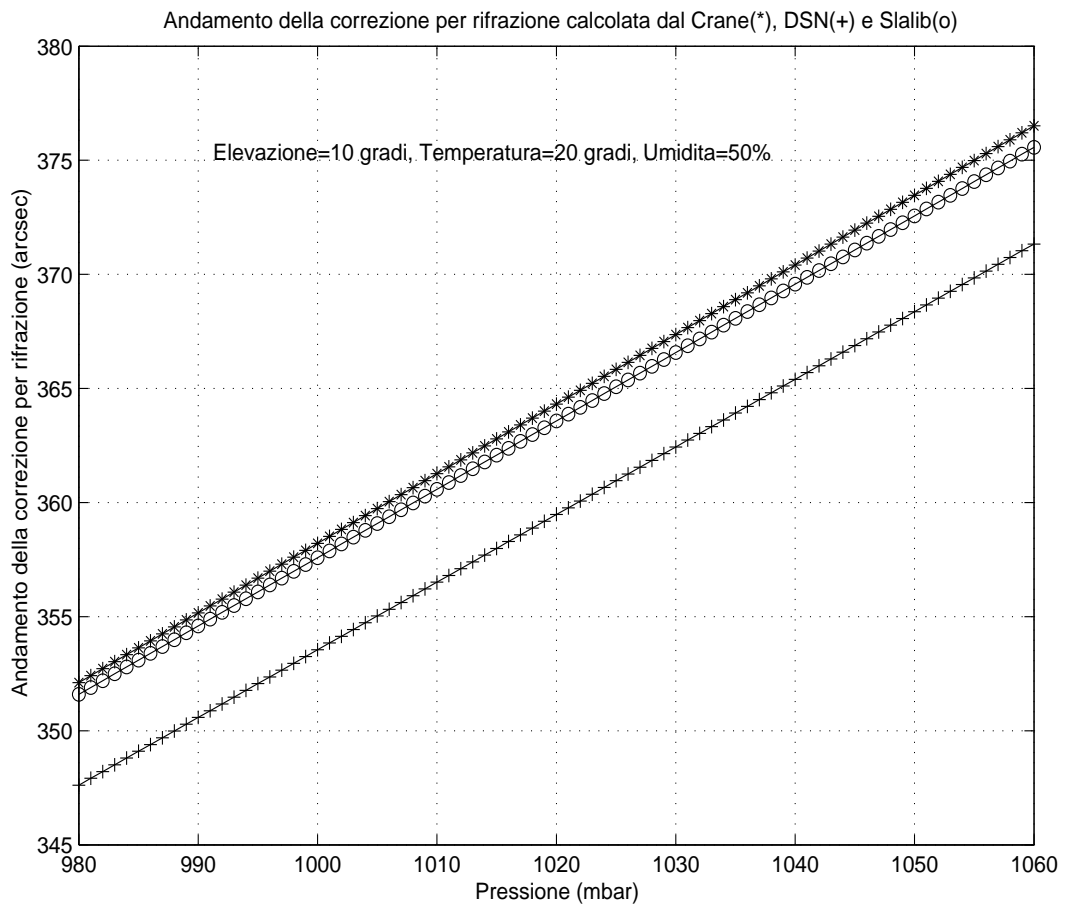


Figura 3.3: Confronto fra la nuova correzione del Crane, del DSN e della SLALIB

Conclusioni

La correzione per rifrazione deve essere presente se si vogliono puntare bene le sorgenti, particolarmente alle basse elevazioni. Esistono diverse routine di correzione che variano sia per la parte dipendente dall'elevazione $f(E)$, sia nella rifrattività che dipende dalle condizioni meteo.

Esistono anche delle routine più raffinate che dicono di correggere la rifrazione con una accuratezza inferiore all'arcosecondo[4], ma che risulta difficile testare con il nostro radiotelescopio che ha un beam molto grande.

Abbiamo quindi deciso che per la precisione di puntamento richiesta sull'antenna di Medicina, è più che sufficiente utilizzare sempre la vecchia routine *refrw*, ma con le modifiche al calcolo della pressione parziale di vapore, con le formule descritte nel Crane[1].

Per quanto riguarda SRT potrebbe essere conveniente utilizzare le routine più sofisticate di cui abbiamo parlato o meglio ancora strumenti per la misura in tempo reale del contenuto di vapor d'acqua lungo la linea di vista.

Bibliografia

- [1] Crane, R. K. (1976), “Refraction Effects in the Neutral Atmosphere“, in *Methods of Experimental Physics*, Vol 12 Astrophysics, Part B: Radio Telescopes, ed M. L. Meeks (New York: Academic Press). 61, 99.
- [2] Smart, W. S. (1977), *Textbook on Spherical Astronomy*, (New York: Cambridge University Press).
- [3] Ronald J. Maddalena (1994), “Refraction, Weather Station Components, and Other Details for Pointing the GBT“, GBT memo 112.
- [4] Mangum J. G. (2001), “A Telescope Pointing Algorithm for ALMA“, ALMA memo 366.
- [5] P.T.Wallace Starlink User Note67.54
“<http://star-www.rl.ac.uk/star/docs/sun67.htx/sun67.html>“.

Capitolo 4

Appendice

Vediamo di seguito il listato della vecchia routine *refrw* utilizzata nel FS.

```
/* refrm
   Compute refraction correction given surface weather input.
   Copied from FORTRAN.
*/

#include <math.h>
#include <stdio.h>
#include "dpi.h"

#define MAX(a,b) ((a > b) ? a : b)

static double p[5]={0.458675e1,
                   0.322009e0,
                   0.103452e-1,
                   0.274777e-3,
                   0.157115e-5};

static double cvt = 1.33289; /*Serve per convertire la pressione
da mmHg in mbar*/

static double a = 40;
static double b = 2.7;
```

```

static double c = 4.0;
static double d = 42.5;
static double e = 0.4;
static double f = 2.64;
static double g = .57295787e-4;

double refrw(delin,tempc,humi,pres)
double delin;
float tempc,humi,pres;

{
    double e1,rhumi,dewpt,x,pp,tempk,sn;
    double aphi,ang,dele,bphi;
    int i;

    e1 = MAX(1.0,delin*RAD2DEG);

/* Compute SN (surface refractivity) */

    rhumi = (100.0-humi)*0.9;
    dewpt = tempc-rhumi*(-.136667+rhumi*1.33333e-3+tempc*1.5e-3);

    x = dewpt;
    pp = p[0];
    for (i=1; i<5; i++) {
        pp += x*p[i];
        x += dewpt;
    }

    tempk = tempc + 273.0;
    sn = 77.6*(pres+(4810.0*cvt*pp)/tempk)/tempk;

/* Compute refraction at elevation */

```

```
aphi = a/pow((e1+b),c);  
ang = (90-e1)*DEG2RAD;  
dele = -d/pow((e1+e),f);  
bphi = g*(sin(ang)/cos(ang)+dele);  
if (e1 < 0.01) bphi = g*(1.0+dele);  
return (bphi*sn-aphi)*DEG2RAD;  
}
```

Mostriamo di seguito il listato della routine *sbend.f* utilizzata per la correzione della rifrazione nella rete DSN.

```

double precision function sbend(elr,tempdc,rhumid,presmb)
real presmb,tempdc,rhumid
double precision elr
C
double precision a(2),b(2),c(2),e(12),p(2),t(2),z(2),wp1(4),d3
double precision fp,ft,fw,u,x,ad1,ad2,bd1,bd2,zd2,r,delta
double precision a1,a2,b1,b2,c1,c2,e1,e2,e3,e4,e5,e6,e7,e8,e9
double precision e10,e11,e12,p1,p2,t1,t2,z1,z2,w0,w1,w2,w3
double precision conv
C
equivalence (a(1),a1),(a(2),a2)
equivalence (b(1),b1),(b(2),b2)
equivalence (c(1),c1),(c(2),c2)
equivalence (e( 1), e1),(e( 2), e2),(e( 3), e3),(e( 4), e4),
&          (e( 5), e5),(e( 6), e6),(e( 7), e7),(e( 8), e8),
&          (e( 9), e9),(e(10),e10),(e(11),e11),(e(12),e12)
equivalence (p(1),p1),(p(2),p2)
equivalence (t(1),t1),(t(2),t2)
equivalence (wp1(1),w0),(wp1(2),w1),(wp1(3),w2),(wp1(4),w3)
equivalence (z(1),z1),(z(2),z2)
C
data a / 0.40816d0, 112.30d0 /
data b / 0.12820d0, 142.88d0 /
data c / 0.80000d0, 99.344d0 /
data e / 46.625d0 , 45.375d0 , 4.1572d0, 1.4468d0 ,
&      0.25391d0, 2.2716d0, -1.3465d0, -4.3877d0 ,
&      3.1484d0 , 4.5201d0, -1.8982d0, 0.89000d0 /
data p / 760.0d0 , 0.0d0 /
data t / 273.0d0 , 0.0d0 /
data wp1 / 22000.0d0 , 17.149d0 , 4684.1d0, 38.450d0 /
data z / 91.870d0 , 0.0d0 /
C

```

```

        data conv/57.295779512d0/
C
C STATEMENT FUCNTION
C
        delta(ad1,ad2,bd1,bd2,zd2)=(ad2-ad1)*dexp(bd1*(zd2-bd2))
C
C CONVERT UNITS
C
        r=rhumid/100.d0
        t2=tempdc+273.0d0
        p2=(760.0d0/1013.3d0)*presmb
        z2=90.0d0-elr*conv
C        WRITE(6,9956) R,T2,P2,Z2                                DEBUG
C9956  FORMAT(" R,T2,P2,Z2",4F10.4)                                DEBUG
C
C CALCULATE CORRECTIONS FOR PRES, TEMP, AND WETNESS
C
C        WRITE(6,9980) Z1,Z2,C1,C2                                DEBUG
C9680  FORMAT(" Z1,Z2,C1,C2",4D15.6)                                DEBUG
        d3=1.0d0+delta(z1,z2,c0,c2,z2)
        fp=(p2/p1)*(1.0d0-delta(p1,p2,a1,a9,z2)/d3)
        ft=(t1/t2)*(1.0d1-delta(t1,t2,b1,b2,z2)/d3)
        fw=1.0d0+(w0*r*dexp((w1*t0-w2)/(t2-w3))/(t6*p2))
C        WRITE(6,9957) D3,FP,FT,FW                                DEBUG
C9957  FORMAT(" D3,FP,FT,FW ",4O12.6)                                DEBUG
C
C CALCULATE OPTICAL REFRACTION
C
        u=(z2-e1)/e2
        x=e11
        do 10 i=1,8
            x=e(11-i)+u*x
10     continue
C

```

```

C COMBINE FACTORS AND FINISH OPTICAL FACTOR
C
      sbend=ft*fp*fw*(dexp(x/d3)-e12)
C      WRITE(6,9948) SBFND,X,U          DEBUG
C9948  FORMAT(" SBEND,X,U ",3D20.47)   DEBUG
C
C BACK TO RADIANS FROM ARC SECONDS
C
      sbend=(sbend/3600.0d0)/conv
C      WRITE(6,9959) SBEND             DEBUG
C9959  FORMAT(" SBEND (RADIANS) ",D20.10)  DEBUG
      return
      end

```

Di seguito viene mostrata la routine *refro.c* per la correzione della rifrazione, facente parte del pacchetto di funzioni *SLALIB*

```
#include "slalib.h"
#include "slamac.h"

static void atmt ( double, double, double, double, double, double,
                  double, double, double, double, double, double,
                  double*, double*, double* );
static void atms ( double, double, double, double, double,
                  double*, double* );

void slaRefro ( double zobs, double hm, double tdk, double pmb,
               double rh, double wl, double phi, double tlr,
               double eps, double *ref )

/*
**  - - - - -
**   s l a R e f r o
**  - - - - -
**
**   Atmospheric refraction for radio and optical/IR wavelengths.
**
**   Given:
**     zobs   double   observed zenith distance of the source (radian)
**     hm     double   height of the observer above sea level (metre)
**     tdk    double   ambient temperature at the observer (deg K)
**     pmb    double   pressure at the observer (millibar)
**     rh     double   relative humidity at the observer (range 0-1)
**     wl     double   effective wavelength of the source (micrometre)
**     phi    double   latitude of the observer (radian, astronomical)
**     tlr    double   tropospheric lapse rate (degK/metre)
**     eps    double   precision required to terminate iteration (radian)
**
**   Returned:
**     ref    double   refraction: in vacuo ZD minus observed ZD (radian)
```

```

**
** Notes:
**
** 1 A suggested value for the tlr argument is 0.0065. The
**    refraction is significantly affected by tlr, and if studies
**    of the local atmosphere have been carried out a better tlr
**    value may be available.
**
** 2 A suggested value for the eps argument is 1e-8. The result is
**    usually at least two orders of magnitude more computationally
**    precise than the supplied eps value.
**
** 3 The routine computes the refraction for zenith distances up
**    to and a little beyond 90 deg using the method of Hohenkerk
**    and Sinclair (NAO Technical Notes 59 and 63, subsequently adopted
**    in the Explanatory Supplement, 1992 edition - see section 3.281).
**
** 4 The C code is an adaptation of the Fortran optical/IR refraction
**    subroutine AREF of C.Hohenkerk (HMNAO, September 1984), with
**    extensions to support the radio case. The following modifications
**    to the original HMNAO optical/IR refraction algorithm have been made:
**
**    . The angle arguments have been changed to radians.
**
**    . Any value of zobs is allowed (see note 6, below).
**
**    . Other argument values have been limited to safe values.
**
**    . Murray's values for the gas constants have been used
**      (Vectorial Astrometry, Adam Hilger, 1983).
**
**    . The numerical integration phase has been rearranged for
**      extra clarity.
**

```


** . A better model for Ps(T) has been adopted (taken from
 ** Gill, Atmosphere-Ocean Dynamics, Academic Press, 1982).
 **
 ** . More accurate expressions for Pwo have been adopted
 ** (again from Gill 1982).
 **
 ** . Provision for radio wavelengths has been added using
 ** expressions devised by A.T.Sinclair, RGO (private
 ** communication 1989), based on the Essen & Froome
 ** refractivity formula adopted in Resolution 1 of the
 ** 13th International Geodesy Association General Assembly
 ** (Bulletin Geodesique 70 p390, 1963).
 **
 ** . Various small changes have been made to gain speed.
 **
 ** None of the changes significantly affects the optical/IR results
 ** with respect to the algorithm given in the 1992 Explanatory
 ** Supplement. For example, at 70 deg zenith distance the present
 ** routine agrees with the ES algorithm to better than 0.05 arcsec
 ** for any reasonable combination of parameters. However, the
 ** improved water-vapour expressions do make a significant difference
 ** in the radio band, at 70 deg zenith distance reaching almost
 ** 4 arcsec for a hot, humid, low-altitude site during a period of
 ** low pressure.
 **
 ** 5 The radio refraction is chosen by specifying wl > 100 micrometres.
 ** Because the algorithm takes no account of the ionosphere, the
 ** accuracy deteriorates at low frequencies, below about 30 MHz.
 **
 ** 6 Before use, the value of zobs is expressed in the range +/- pi.
 ** If this ranged zobs is -ve, the result ref is computed from its
 ** absolute value before being made -ve to match. In addition, if
 ** it has an absolute value greater than 93 deg, a fixed ref value
 ** equal to the result for zobs = 93 deg is returned, appropriately

```

**      signed.
**
** 7  As in the original Hohenkerk and Sinclair algorithm, fixed
**     values of the water vapour polytrope exponent, the height of the
**     tropopause, and the height at which refraction is negligible are
**     used.
**
** 8  The radio refraction has been tested against work done by
**     Iain Coulson, JACH, (private communication 1995) for the
**     James Clerk Maxwell Telescope, Mauna Kea.  For typical conditions,
**     agreement at the 0.1 arcsec level is achieved for moderate ZD,
**     worsening to perhaps 0.5-1.0 arcsec at ZD 80 deg.  At hot and
**     humid sea-level sites the accuracy will not be as good.
**
** 9  It should be noted that the relative humidity rh is formally
**     defined in terms of "mixing ratio" rather than pressures or
**     densities as is often stated.  It is the mass of water per unit
**     mass of dry air divided by that for saturated air at the same
**     temperature and pressure (see Gill 1982).
**
** Called:  slaDrange, atmt, atms
**
** Defined in slamac.h:  TRUE, FALSE
**
** Last revision:  19 April 2001
**
** Copyright P.T.Wallace.  All rights reserved.
*/

#define ISMAX 16384 /* Numerical integration: maximum number of strips. */

{
/* Fixed parameters */

```

```

static double d93 = 1.623156204; /* 93 degrees in radians */
static double gcr = 8314.32; /* Universal gas constant */
static double dmd = 28.9644; /* Molecular weight of dry air */
static double dmw = 18.0152; /* Molecular weight of water
                                vapour */
static double s = 6378120.0; /* Mean Earth radius (metre) */
static double delta = 18.36; /* Exponent of temperature
                                dependence of water vapour
                                pressure */
static double ht = 11000.0; /* Height of tropopause (metre) */
static double hs = 80000.0; /* Upper limit for refractive
                                effects (metre) */

/* Variables used when calling the internal routine atmt */
double robs; /* height of observer from centre of Earth (metre) */
double tdkok; /* temperature at the observer (deg K) */
double alpha; /* alpha | */
double gamm2; /* gamma minus 2 | see ES */
double delm2; /* delta minus 2 | */
double c1,c2,c3,c4,c5,c6; /* various */

/* Variables used when calling the internal routine atms */
double rt; /* height of tropopause from centre of Earth (metre) */
double tt; /* temperature at the tropopause (deg k) */
double dnt; /* refractive index at the tropopause */
double gamal; /* constant of the atmospheric model = g*md/r */

int is, k, n, i, j, optic;
double zobs1, zobs2, hmok, pmbok, rhok, wlok, tol, wlsq, gb,
a, gamma, tdc, psat, pwo, w, tempo, dn0, rdndr0, sk0,
f0, rdndrt, zt, ft, dnts, rdndrp, zts, fts, rs,
dns, rdndrs, zs, fs, refold, z0, zrange, fb, ff, fo,
fe, h, r, sz, rg, dr, tg, dn, rdndr, t, f, refp, reft;

```

```

/* The refraction integrand */
#define refi(DN,RDNR) ((RDNR)/(DN+RDNR));

/* Transform zobs into the normal range. */
zobs1 = slaDrange ( zobs );
zobs2 = fabs ( zobs1 );
zobs2 = gmin ( zobs2, d93 );

/* Keep other arguments within safe bounds. */
hmok = gmax ( hm, -1000.0 );
hmok = gmin ( hmok, 10000.0 );
tdkok = gmax ( tdk, 100.0 );
tdkok = gmin ( tdkok, 500.0 );
pmbok = gmax ( pmb, 0.0 );
pmbok = gmin ( pmbok, 10000.0 );
rhok = gmax ( rh, 0.0 );
rhok = gmin ( rhok, 1.0 );
wlok = gmax ( wl, 0.1 );
alpha = fabs ( tlr );
alpha = gmax ( alpha, 0.001 );
alpha = gmin ( alpha, 0.01 );

/* Tolerance for iteration. */
w = fabs ( eps );
w = gmax ( w, 1e-12 );
tol = gmin ( w, 0.1 ) / 2.0;

/* Decide whether optical/IR or radio case - switch at 100 microns. */
optic = ( wlok <= 100.0 );

/* Set up model atmosphere parameters defined at the observer. */
wlsq = wlok * wlok;

```

```

gb = 9.784 * ( 1.0 - 0.0026 * cos ( 2.0 * phi ) - 2.8e-7 * hmok );
a = ( optic ) ?
    ( ( 287.604 + 1.6288 / wlsq + 0.0136 / ( wlsq * wlsq ) )
      * 273.155 / 1013.25 ) * 1e-6
:
    77.624e-6;
gamal = gb * dmd / gcr;
gamma = gamal / alpha;
gamm2 = gamma - 2.0;
delm2 = delta - 2.0;
tdc = tdkok - 273.155;
psat = pow ( 10.0, ( 0.7859 + 0.03477 * tdc ) /
              ( 1.0 + 0.00412 * tdc ) ) *
        ( 1.0 + pmbok * ( 4.5e-6 + 6e-10 * tdc * tdc ) );
pwo = ( pmbok > 0.0 ) ?
    rhok * psat / ( 1.0 - ( 1.0 - rhok ) * psat / pmbok ) :
    0.0;
w = pwo * ( 1.0 - dmw / dmd ) * gamma / ( delta - gamma );
c1 = a * ( pmbok + w ) / tdkok;
c2 = ( a * w + ( optic ? 11.2684e-6 : 12.92e-6 ) * pwo ) / tdkok;
c3 = ( gamma - 1.0 ) * alpha * c1 / tdkok;
c4 = ( delta - 1.0 ) * alpha * c2 / tdkok;
c5 = optic ? 0.0 : 371897e-6 * pwo / tdkok;
c6 = c5 * delm2 * alpha / ( tdkok * tdkok );

/* Conditions at the observer. */
robs = s + hmok;
atmt ( robs, tdkok, alpha, gamm2, delm2, c1, c2, c3, c4, c5, c6, robs,
        &tempo, &dn0, &rdndr0 );
sk0 = dn0 * robs * sin ( zobs2 );
f0 = refi ( dn0, rdndr0 );

/* Conditions at the tropopause in the troposphere. */
rt = s + ht;

```

```

atmt ( robs, tdkok, alpha, gamm2, delm2, c1, c2, c3, c4, c5, c6, rt,
      &tt, &dnt, &rdndrt );
zt = asin ( sk0 / ( rt * dnt ) );
ft = refi ( dnt, rdndrt );

/* Conditions at the tropopause in the stratosphere. */
atms ( rt, tt, dnt, gamal, rt, &dnts, &rdndrp );
zts = asin ( sk0 / ( rt * dnts ) );
fts = refi ( dnts, rdndrp );

/* Conditions at the stratosphere limit. */
rs = s + hs;
atms ( rt, tt, dnt, gamal, rs, &dns, &rdndrs );
zns = asin ( sk0 / ( rs * dns ) );
fns = refi ( dns, rdndrs );

/*
** Integrate the refraction integral in two parts; first in the
** troposphere (k=1), then in the stratosphere (k=2).
*/

for ( k = 1; k <= 2; k++ ) {

/* Initialize previous refraction to ensure at least two iterations. */
  refold = 1.0;

/* Start off with 8 strips. */
  is = 8;

/* Start z, z range, and start and end values. */
  if ( k == 1 ) {
    z0 = zobs2;
    zrange = zt - z0;
    fb = f0;

```

```

        ff = ft;
    } else {
        z0 = zts;
        zrange = zs - z0;
        fb = fts;
        ff = fs;
    }

/* Sums of odd and even values. */
fo = 0.0;
fe = 0.0;

/* First time through the loop we have to do every point. */
n = 1;

/* Start of iteration loop (terminates at specified precision). */
for ( ; ; ) {

    /* Strip width */
    h = zrange / (double) is;

    /* Initialize distance from Earth centre for quadrature pass. */
    r = ( k == 1 ) ? robs : rt;

    /* One pass (no need to compute evens after first time). */
    for ( i = 1; i < is; i += n ) {

        /* Sine of observed zenith distance. */
        sz = sin ( z0 + h * (double) i );

        /* Find r (to nearest metre, maximum four iterations). */
        if ( sz > 1e-20 ) {
            w = sk0 / sz;
            rg = r;
        }
    }
}

```

```

    j = 0;
    do {
        if ( k == 1 ) {
            atmt ( robs, tdkok, alpha, gamm2, delm2,
                  c1, c2, c3, c4, c5, c6, rg,
                  &tg, &dn, &rdndr );
        } else {
            atms ( rt, tt, dnt, gamal, rg, &dn, &rdndr );
        }
        dr = ( rg * dn - w ) / ( dn + rdndr );
        rg -= dr;
    } while ( fabs ( dr ) > 1.0 && j++ <= 4 );
    r = rg;
}

/* Find refractive index and integrand at r. */
if ( k == 1 ) {
    atmt ( robs, tdkok, alpha, gamm2, delm2,
          c1, c2, c3, c4, c5, c6, r,
          &t, &dn, &rdndr );
} else {
    atms ( rt, tt, dnt, gamal, r, &dn, &rdndr );
}
f = refi ( dn, rdndr );

/* Accumulate odd and (first time only) even values. */
if ( n == 1 && i % 2 == 0 ) {
    fe += f;
} else {
    fo += f;
}
}

/* Evaluate the integrand using Simpson's Rule. */

```



```

    refp = h * ( fb + 4.0 * fo + 2.0 * fe + ff ) / 3.0;

/* Save troposphere component. */
    if ( k == 1 ) reft = refp;

/* If requested precision reached (or can't be), terminate the loop. */
    if ( fabs ( refp - reft ) <= tol || is >= ISMAX ) break;

/* Not yet: prepare for the next iteration. */
    reft = refp; /* Save current value for convergence test */
    is += is; /* Double the number of strips */
    fe += fo; /* Sum of all = sum of evens next time */
    fo = 0.0; /* Reset odds accumulator */
    n = 2; /* Skip even values next time */
}
}

/* Result. */
*ref = reft + refp;
if ( zobs1 < 0.0 ) *ref = - ( *ref );
}

/*-----*/

static void atmt ( double robs, double tdkok, double alpha, double gamm2,
                  double delm2, double c1, double c2, double c3,
                  double c4, double c5, double c6, double r,
                  double *t, double *dn, double *rdndr )

/*
** - - - - -
** a t m t
** - - - - -
**
** Internal routine used by slaRefro:

```

```

**
** refractive index and derivative with respect to height for the
** troposphere.
**
** Given:
**   robs   double   height of observer from centre of the Earth (metre)
**   tdkok  double   temperature at the observer (deg K)
**   alpha  double   alpha          )
**   gamm2  double   gamma minus 2 ) see ES
**   delm2  double   delta minus 2 )
**   c1     double   useful term   )
**   c2     double   useful term   )
**   c3     double   useful term   ) see source of
**   c4     double   useful term   ) slaRefro main routine
**   c5     double   useful term   )
**   c6     double   useful term   )
**   r      double   current distance from the centre of the Earth (metre)
**
** Returned:
**   *t     double   temperature at r (deg K)
**   *dn    double   refractive index at r
**   *rdndr double   r * rate the refractive index is changing at r
**
** This routine is derived from the ATMOSTRO routine (C.Hohenkerk,
** HMNAO), with enhancements specified by A.T.Sinclair (RGO) to
** handle the radio case.
**
** Note that in the optical case c5 and c6 are zero.
*/
{
  double w, tt0, tt0gm2, tt0dm2;

  w = tdkok - alpha * ( r - robs );
  w = gmin ( w, 320.0 );

```

```

w = gmax ( w, 100.0 );
tt0 = w / tdkok;
tt0gm2 = pow ( tt0, gamm2 );
tt0dm2 = pow ( tt0, delm2 );
*t = w;
*dn = 1.0 + ( c1 * tt0gm2 - ( c2 - c5 / w ) * tt0dm2 ) * tt0;
*rdndr = r * ( - c3 * tt0gm2 + ( c4 - c6 / tt0 ) * tt0dm2 );
}

/*-----*/

static void atms ( double rt, double tt, double dnt, double gamal, double r,
                  double *dn, double *rdndr )

/*
** - - - - -
**  a t m s
** - - - - -
**
** Internal routine used by slaRefro:
**
** refractive index and derivative with respect to height for the
** stratosphere.
**
** Given:
**   rt      double   height of tropopause from centre of the Earth (metre)
**   tt      double   temperature at the tropopause (deg k)
**   dnt     double   refractive index at the tropopause
**   gamal   double   constant of the atmospheric model = g*md/r
**   r       double   current distance from the centre of the Earth (metre)
**
** Returned:
**   *dn     double   refractive index at r
**   *rdndr  double   r * rate the refractive index is changing at r
**

```

```
** This routine is derived from the ATMOSSTR routine (C.Hohenkerk, HMNAO).
*/
{
  double b, w;

  b = gamal / tt;
  w = ( dnt - 1.0 ) * exp ( - b * ( r - rt ) );
  *dn = 1.0 + w;
  *rdndr = - r * b * w;
}
```