GPS 응용

Satellite Navigation Lab
Inha University

2025. 3.27.

설 윤 환

학습 목표

- GIM을 이용한 전리층 지연 보정
  1. IONEX format
  2. ReadGIM.p 함수를 사용하여 MATLAB에 불러오기
  3. IPP 산출
  4. 시간 및 공간 보간(GetVTECgim.p, spline.p)
  5. VTEC -> STEC

- GPT/GMF를 이용한 대류권 지연 보정
  1. gpt_v1.m 함수를 사용하여 특정 지점, 시점에서의 대기압(hpa) 산출
  2. 대기압을 이용하여 ZHD 산출
  3. gmf_f_hu.m 함수를 사용하여 ZHD -> SHD

IONEX

- GIM: Global Ionosphere Map
  - IONEX: **Ion**osphere **ex**change format
  - https://cddis.nasa.gov/archive/gnss/products/ionex/2024/001/

year doy

The ionospheric products since GPS week 2238 (November 26, 2022), are in transition to the IGS long product filename convention.
The available products are listed in the table below together with their previous short names and new long names.

(reference) https://igs.org/products/#ionosphere

File type	Old short name	New long name
Final combined IONEX	igsg**ddd**0.**yy**i.Z	IGS0OPSFIN_**yyyyddd**0000_01D_02H_GIM.INX.gz
Rapid combined IONEX	igrg**ddd**0.**yy**i.Z	IGS0OPSRAP_**yyyyddd**0000_01D_02H_GIM.INX.gz

AC

- **ddd**: day of year [001–366]
- **yy**: 2-digit year
- **yyyy**: 4-digit year

IONEX

- Epoch of Current Map
  - Y/M/D/h/m/s
- LAT/LON1/LON2/DLON/H
  - LAT
  - LON1~LON2까지 DLON 간격
  - Ionosphere height (km) = **450**
- VTEC
  - Vertical Total Electron Content
  - 단위: 10 TECU
- Time Interval
  - 0~24h, 2h interval → 13 Epochs (case by case)

```
2015     2     1     0     0     0                        EPOCH OF CURRENT MAP
  87.5-180.0 180.0   5.0 450.0                            LAT/LON1/LON2/DLON/H
 137  138  140  141  143  145  146  146  147  148  148  148  148  148  147  146
 145  144  142  140  138  135  133  131  128  126  123  120  117  115  112  109
 107  105  103  101  100   98   97   96   96   95   95   95   95   95   96   96
  97   98   99  100  101  102  104  105  107  109  110  112  114  115  117  119
 121  123  125  128  129  131  133  135  137
  85.0-180.0 180.0   5.0 450.0                            LAT/LON1/LON2/DLON/H
 150  154  157  160  164  167  169  172  174  176  178  179  179  178  178  177
 175  173  170  166  162  158  153  148  142  137  131  125  119  114  109  104
 100   96   93   90   88   86   85   84   84   83   84   85   85   86   87   88
  89   91   92   94   95   98   99  101  104  106  108  111  113  116  119  122
 125  128  131  134  137  140  144  147  150
  82.5-180.0 180.0   5.0 450.0                            LAT/LON1/LON2/DLON/H
 158  163  168  173  179  184  190  194  198  201  204  206  206  207  206  205
 202  199  195  191  185  179  173  166  158  150  142  134  125  118  110  104
  98   93   89   86   84   83   83   84   85   86   87   89   91   92   93   95
  96   97   98   99  101  102  103  105  107  109  111  113  116  118  121  124
 127  130  133  137  141  145  149  154  158
  80.0-180.0 180.0   5.0 450.0                            LAT/LON1/LON2/DLON/H
 163  169  177  184  193  200  208  214  221  225  228  229  229  228  226  224
 221  217  212  208  202  196  190  182  174  164  155  144  135  125  116  108
 101   95   91   89   89   90   92   94   97  100  103  106  109  110  111  112
 113  113  113  113  112  112  112  112  113  114  115  116  118  120  123  125
 128  131  134  138  142  146  152  157  163
```

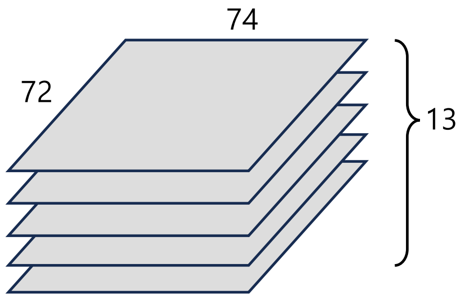
IONEX

- [Lat, Lon, TEC] = ReadGIM(filename)
  - Example:

    ```
    [Lat,Lon,TEC]=ReadGIM('JPL0OPSFIN_20240010000_01D_02H_GIM.INX');
    ```

  - input: Filename of IONEX

  - output
    - Lat: TEC 값이 제공된 위도 List
    - Lon: TEC 값이 제공된 경도 List
    - TEC: 시간/위도/경도 별 TEC List



Lat		1×71 double
Lon		1×73 double
TEC		72×74×13 double

GPS week second

Longitude

VTEC

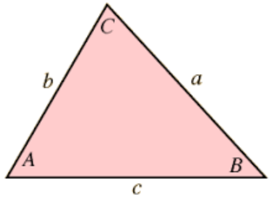
	1	2	3	4	5	6	7	8	9	10	11
1	86400	-180	-175	-170	-165	-160	-155	-150	-145	-140	-135
2	87.5000	7.8000	7.8000	7.8000	7.8000	7.8000	7.8000	7.8000	7.8000	7.8000	7.8000
3	85	7.9000	8	8	8	8	8.1000	8.1000	8.2000	8.2000	8.2000
4	82.5000	8.2000	8.2000	8.3000	8.3000	8.4000	8.5000	8.5000	8.6000	8.6000	8.7000
5	80	8.4000	8.5000	8.6000	8.6000	8.7000	8.7000	8.9000	8.9000	9	9.1000
6	77.5000	8.6000	8.6000	8.6000	8.7000	8.8000	8.9000	9	9.1000	9.2000	9.3000
7	75	8.5000	8.5000	8.6000	8.7000	8.7000	8.9000	9	9.1000	9.3000	9.5000
8	72.5000	8.5000	8.5000	8.5000	8.6000	8.7000	8.8000	8.9000	9.1000	9.4000	9.6000
9	70	8.5000	8.5000	8.5000	8.6000	8.6000	8.7000	8.8000	9	9.2000	9.4000
10	67.5000	8.5000	8.5000	8.5000	8.6000	8.6000	8.6000	8.7000	8.9000	8.9000	9.1000
11	65	8.6000	8.7000	8.7000	8.7000	8.7000	8.7000	8.7000	8.8000	8.8000	8.9000
12	62.5000	8.9000	8.9000	9	8.9000	8.9000	8.8000	8.8000	8.8000	8.7000	8.8000
13	60	9.3000	9.3000	9.3000	9.3000	9.2000	9.1000	9.1000	9	8.9000	8.8000
14	57.5000	9.8000	9.7000	9.8000	9.8000	9.7000	9.6000	9.4000	9.3000	9.1000	8.9000
15	55	10.3000	10.3000	10.4000	10.4000	10.3000	10.1000	10	9.8000	9.5000	9.2000
16	52.5000	10.9000	11	11	11	11	10.8000	10.6000	10.3000	10	9.7000
17	50	11.6000	11.7000	11.6000	11.7000	11.7000	11.5000	11.3000	11	10.6000	10.2000
18	47.5000	12.4000	12.4000	12.3000	12.2000	12.2000	12.2000	11.9000	11.5000	11.1000	10.7000
19	45	12.9000	13	13	12.8000	12.7000	12.7000	12.4000	12.1000	11.7000	11.2000

Latitude

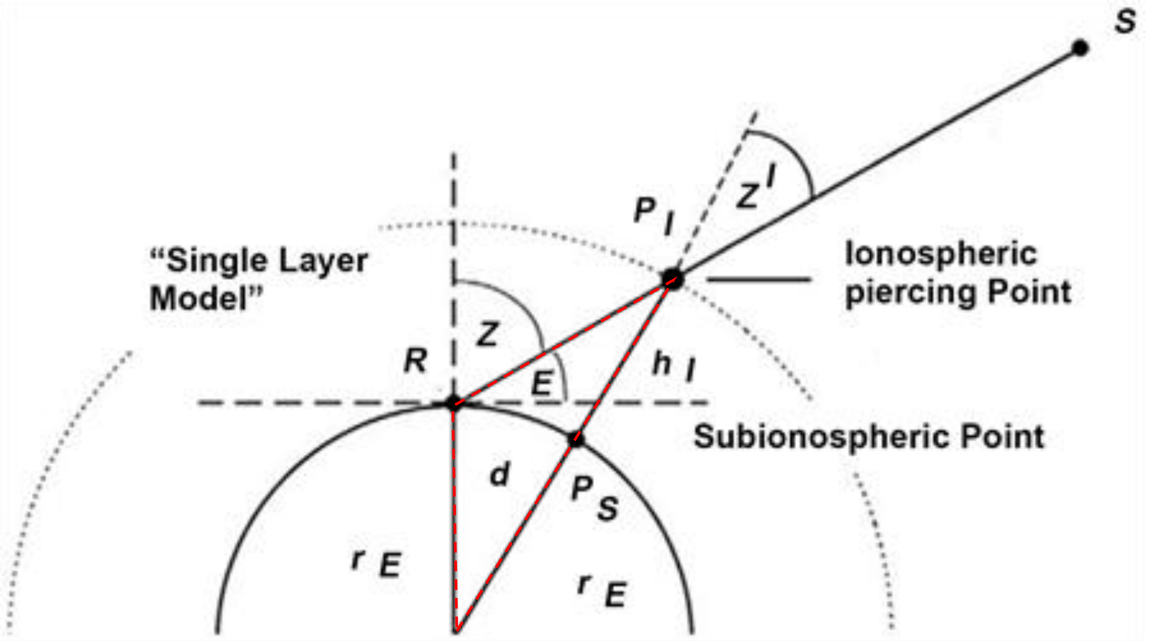
IPP(Ionospheric Pierce Point)

- How to get IPP
  - 1) z' 계산: sin 법칙
    - $\frac{\sin(180-Z)}{R_E+h} = \frac{\sin(Z')}{R_E}$ ➔ $\frac{\sin(Z)}{R_E+h} = \frac{\sin(Z')}{R_E}$
  - 2) IPP까지 거리 계산: cos 법칙

$$c^2 = a^2 + b^2 - 2ab\cos C$$

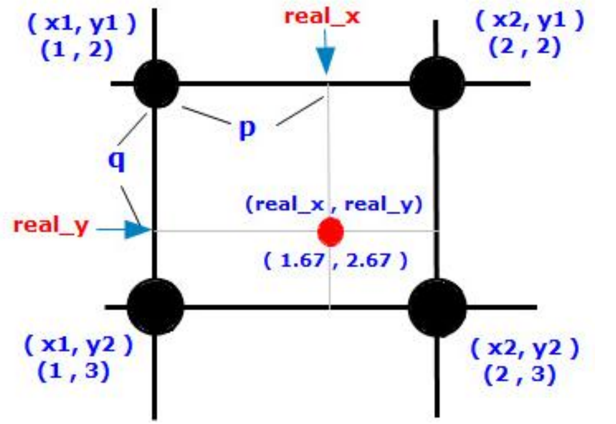


  - 2) 수신기 ➔ IPP 벡터 계산
    - IPP까지 거리 × $(\frac{\vec{\rho}}{|\rho|})$
  - 3) IPP: 수신기 벡터(vec_site) + 수신기 ➔ IPP 벡터



Sapatial Interpolation

- VTEC 공간 보간
  - 1) 인접한 4개의 격자 선정 및 VTEC 추출
  - 2) 공간 보간
    - Linear, Spline, Lagrange, …



$$E(\lambda_0 + p\,\Delta\lambda, \beta_0 + q\,\Delta\beta) = (1-p)\,(1-q)\,E_{0,0} + p\,(1-q)\,E_{1,0} + q\,(1-p)\,E_{0,1} + p\,q\,E_{1,1},$$

where $0 \le p < 1$ and $0 \le q < 1$. $\Delta\lambda$ and $\Delta\beta$ denote the grid widths in longitude and latitude.

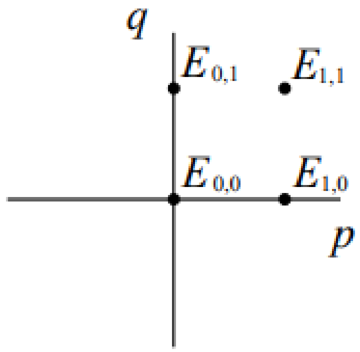
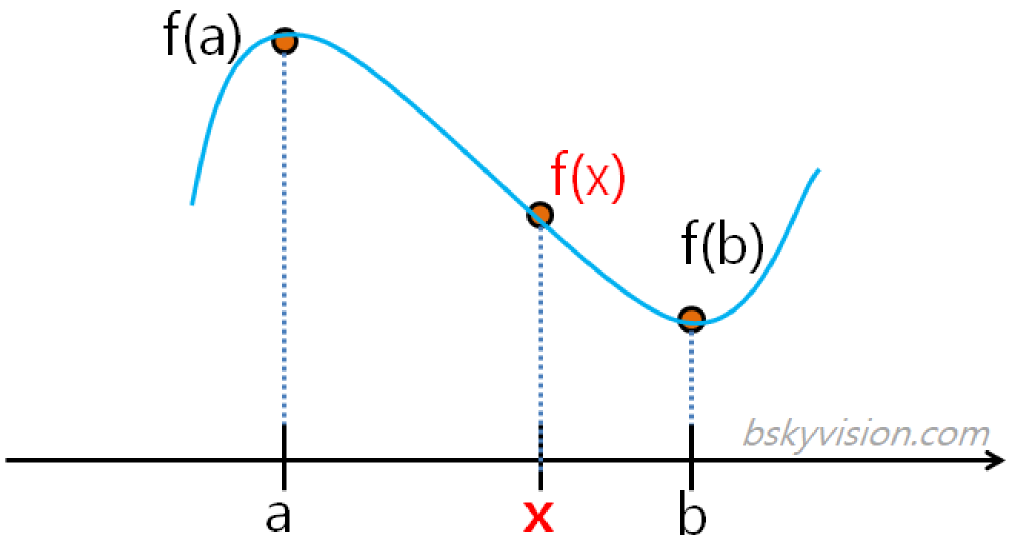


Figure 1: Bivariate interpolation using the nearest 4 TEC values $E_{i,j}$

Time Interpolation

- VTEC 시간 보간
  - VTEC은 2시간 간격 제공 → 시간 보간 필요
    - Linear, Spline, Lagrange, …



- Simply take the nearest TEC map $E_i = E(T_i)$ at epoch T_i:

$$E(\beta, \lambda, t) = E_i(\beta, \lambda), \tag{1}$$

where $|t - T_i| = \min$.

- Interpolate between consecutive TEC maps $E_i = E(T_i)$ and $E_{i+1} = E(T_{i+1})$:

$$E(\beta, \lambda, t) = \frac{T_{i+1} - t}{T_{i+1} - T_i} E_i(\beta, \lambda) + \frac{t - T_i}{T_{i+1} - T_i} E_{i+1}(\beta, \lambda), \tag{2}$$

where $T_i \le t < T_{i+1}$.

- Interpolate between consecutive *rotated* TEC maps:

$$E(\beta, \lambda, t) = \frac{T_{i+1} - t}{T_{i+1} - T_i} E_i(\beta, \lambda_i') + \frac{t - T_i}{T_{i+1} - T_i} E_{i+1}(\beta, \lambda_{i+1}'), \tag{3}$$

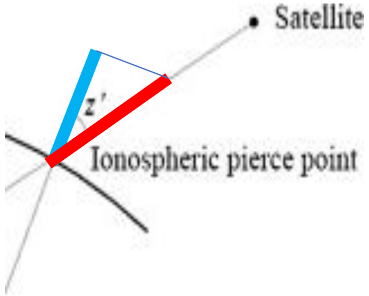
```
VTECgim = GetVTECgim(Lat,Lon,TEC,lat_ipp,lon_ipp,tt);
```

VTEC ➔ STEC

- OF(Obliquity Factor)
  - $OF = \frac{1}{\cos(z')}$

  - STEC = OF * VTEC = $\frac{\text{VTEC}}{\cos(z')}$

  - Ionosphere delay = STEC * 0.162372m (GPS L1)
    - L1: 1575.42 MHz = 1575.42 * 10^6 Hz
    - 1 STEC → 0.16m Error (L1)
    - 1 TECU = 10^{16} electrons / m^2



$$\delta I_g = \frac{40.3}{f^2} TEC \qquad \delta I_\phi = -\frac{40.3}{f^2} TEC$$

IonoGIM.m

```
[dIono,lat_ipp,lon_ipp] = IonoGIM(Lat,Lon,TEC,vec_sat,vec_site,tt)
```

[DO]

1. IPP 산출
2. 시간 및 공간 보간(GetVTECgim.p, spline.p)
3. VTEC -> STEC
4. STEC -> 총 지연량(meters)

[RETURN]

1. dIono : 전리층 지연량
2. lat_ipp : ipp의 위도
3. lon_ipp : ipp의 경도

Global Pressure Temperature -> ZHD

- 사용법

```
[pres,temp,undu] = gpt_v1 (dmjd,dlat,dlon,dhgt)
```

- 입력

dmjd : modified julian date

dlat : latitude in radians

dlon : longitude in radians

dhgt : ellipsoidal height in m

- 반환

pres: pressure in hPa
temp: temperature in Celsius
undu: Geoid undulation in m (from a 9x9 EGM based model)

$$ZHD = \frac{0.0022767 \cdot p}{1 - 0.00266 \cdot cos2\varphi - 0.00028 \cdot h}$$

- φ: Ellipsoidal latitude
- h: Surface height above the ellipsoid in [km]
- p: Total Surface pressure in [hPa]

Global Mapping function

$$SPD(\epsilon) = m_h(\epsilon) \cdot ZHD + m_W(\epsilon) \cdot ZWD \qquad (1)$$

- SPD(ϵ) $=$ slant path delay [m] along elevation angle ϵ
- ZHD = zenith hydrostatic delay [m]
- ZWD = zenith wet delay [m]
- $m_h(\epsilon), m_W(\epsilon)$ are the mapping functions of the hydrostatic and wet path delay along elevation angle ε

```
[gmfh,~] = gmf_f_hu(mJD,dlat,dlon,dhgt,zd);
```

Input:
dmjd : modified julian date
dlat: latitude in radians
dlon: longitude in radians
dhgt: height in m
zd: zenith degree in radians

Output:
gmfh: hydrostatic mapping function
gmfw: wet mapping function