

# 射电干涉测量原理与基线转换公式

## 1. 基本原理

### 1.1 干涉测量核心方程

复可见度表示为：

$$V(u, v, w) = \iint I(l, m) e^{-2\pi i(ul+vm+w(n-1))} \frac{dl dm}{n}$$

其中：

- $(u, v, w)$ ：基线坐标（波长单位）
- $(l, m, n)$ ：天空方向余弦 ( $n = \sqrt{1 - l^2 - m^2}$ )
- $I(l, m)$ ：天空亮度分布

### 1.2 相位延迟模型

对于点源 (R.A.=0°, Dec=80°)：

$$\phi = 2\pi \mathbf{B} \cdot \mathbf{s} / \lambda$$

其中基线矢量  $\mathbf{B}$  和方向矢量  $\mathbf{s}$  的几何关系决定相位变化。

## 2. 基线转换公式

### 2.1 地固坐标系 → 惯性坐标系

基线矢量从地固系到惯性系的转换：

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \sin H & \cos H & 0 \\ -\sin \delta \cos H & \sin \delta \sin H & \cos \delta \\ \cos \delta \cos H & -\cos \delta \sin H & \sin \delta \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}$$

### 2.2 东西向基线简化

对于纯东西向基线 ( $B_y = B_z = 0$ )：

$$\begin{cases} u = \frac{B_x}{\lambda} \sin H \\ v = -\frac{B_x}{\lambda} \sin \delta \cos H \\ w = \frac{B_x}{\lambda} \cos \delta \cos H \end{cases}$$

## 2.3 北天极相位中心特例

当相位中心为北天极时：

$$w \equiv 0 \quad (\text{自动满足})$$

```
In [4]: import numpy as np
import matplotlib.pyplot as plt

# 常量设置
c = 3e8 # 光速 (m/s)
B = 100 # 基线长度 (m)
dec = np.radians(80) # 赤纬 (rad)

# 频率设置 (6144个信道)
num_channels = 6144
freqs = 50e6 + (200e6/8192)*np.arange(num_channels) # 频率数组 (Hz)
wavelengths = c / freqs # 波长数组 (m)

# 时间采样 (24小时, 1分钟分辨率)
time_hours = np.linspace(0, 24, 1440)
HA = np.radians(15*(time_hours-12)) # 时角 (rad)

# 计算uv坐标 (w=0)
u = (B * np.sin(HA))[:, None] / wavelengths
v = -(B * np.sin(dec) * np.cos(HA))[:, None] / wavelengths

# 计算可见度 (相位phi)
phi = 2*np.pi * (B * np.cos(dec) * np.cos(HA))[:, None] * freqs / c
V_real = np.cos(phi)
V_imag = -np.sin(phi)

# 绘制u-v覆盖
plt.figure(figsize=(10, 8))
for i in range(0, num_channels, 300): # 每300个信道画一次
    plt.plot(u[:, i], v[:, i], 'b-', alpha=0.1)
plt.xlabel('u (wavelengths)')
plt.ylabel('v (wavelengths)')
plt.title('u-v Coverage for E-W Baseline (Dec=80°)')
plt.grid()
plt.show()

# 绘制可见度实部和虚部
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.pcolormesh(freqs/1e6, time_hours, V_real, shading='auto', cmap='RdBu', v
plt.colorbar(label='Real Part')
plt.ylabel('Time (hours)')
plt.title('Visibility Real Part')

plt.subplot(2, 1, 2)
plt.pcolormesh(freqs/1e6, time_hours, V_imag, shading='auto', cmap='RdBu', v
plt.colorbar(label='Imaginary Part')
plt.xlabel('Frequency (MHz)')
```

```
plt.ylabel('Time (hours)')
plt.title('Visibility Imaginary Part')

plt.tight_layout()
plt.show()
```

