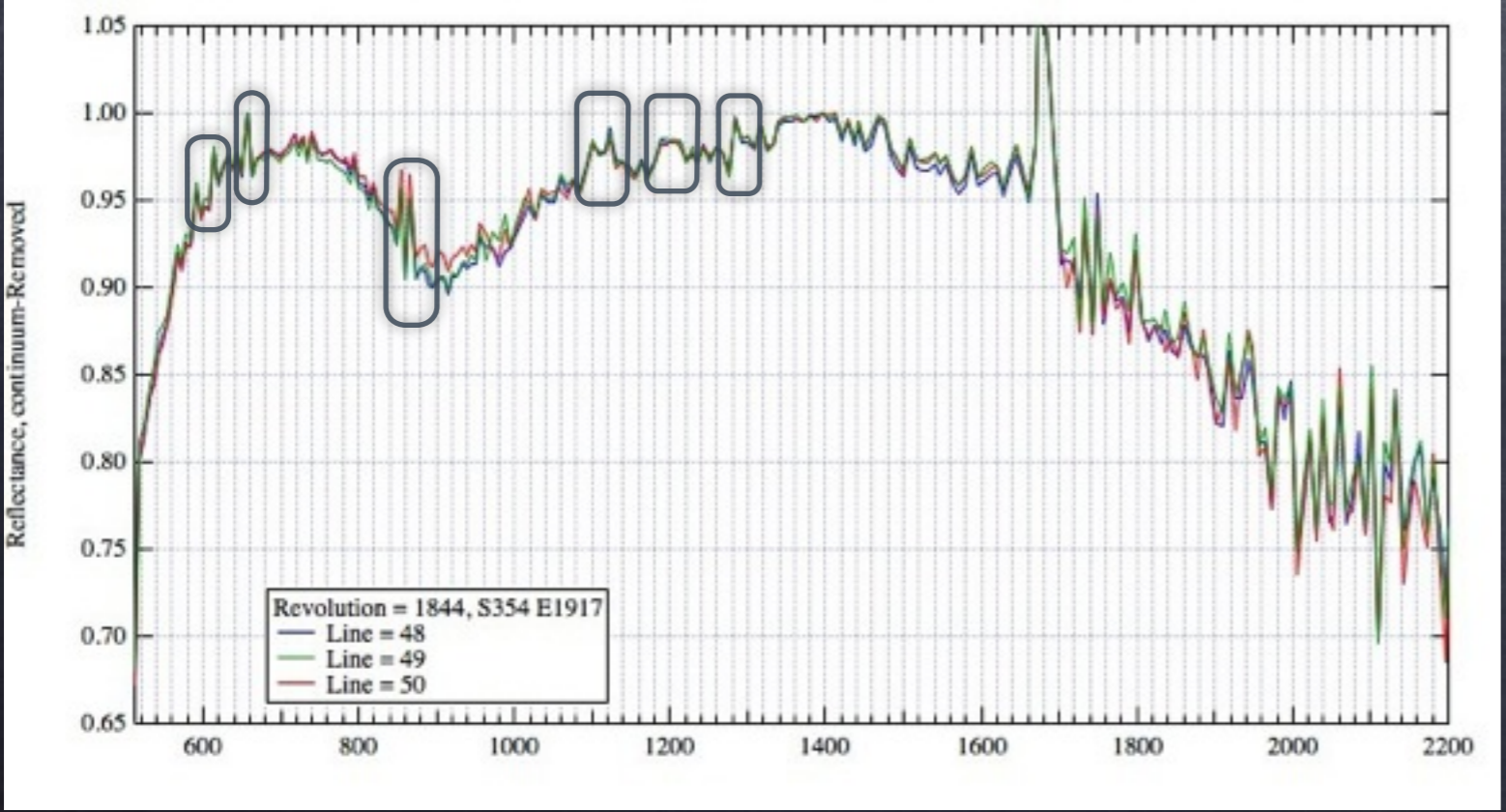
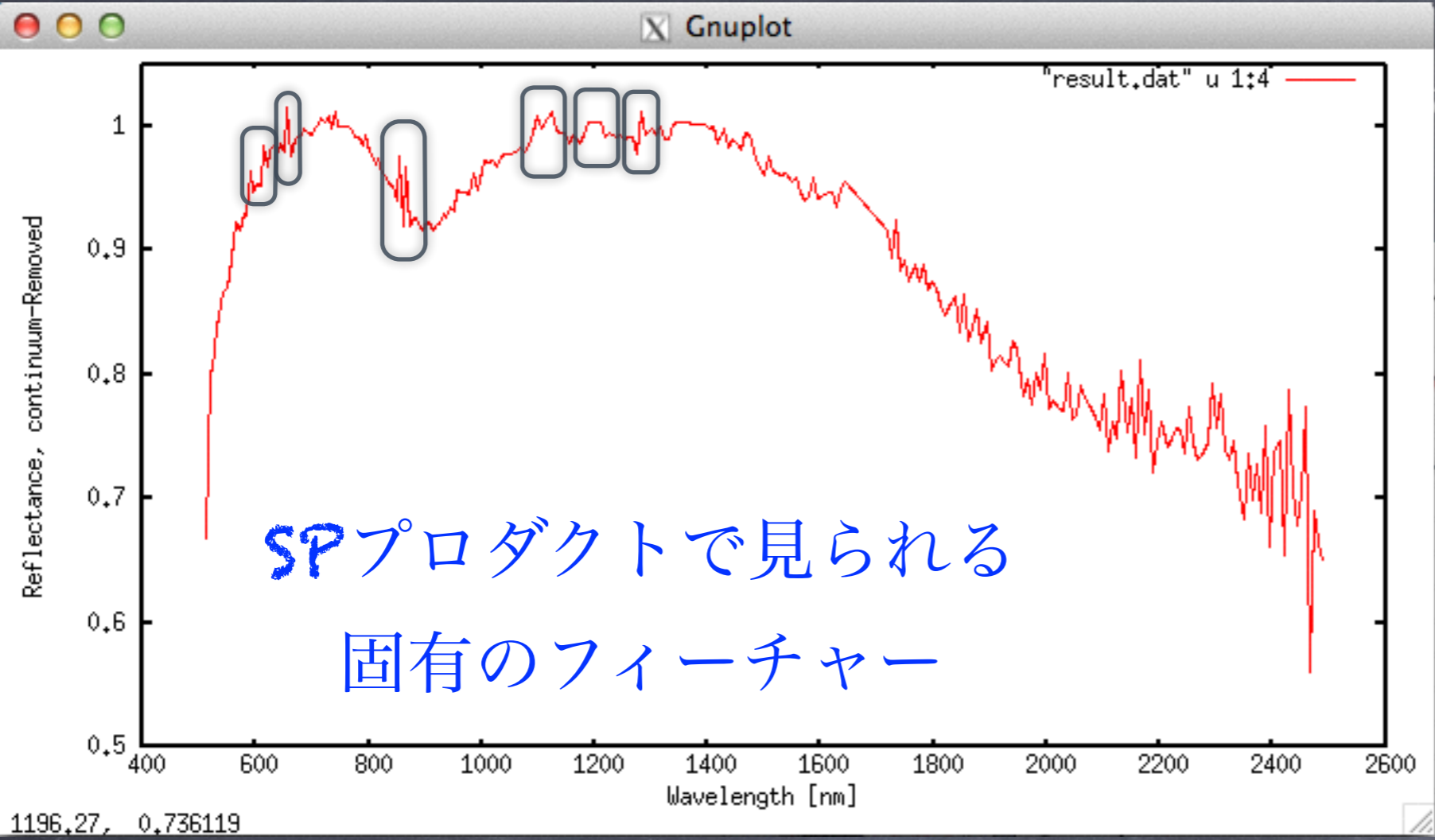


# 4. 課題演習 (応用編)

## 4. 課題演習 (応用編)

かぐや/可視-近赤外連続分光データの基礎校正  
反射率補正係数の作成

# 4.1 反射率補正係数とは



観測場所や、観測時期や  
条件によらず普遍的に見  
られる

## 4.1 反射率補正係数とは

(1) オリジナル値から暗時値を差し引いた値

→  $S$ 値

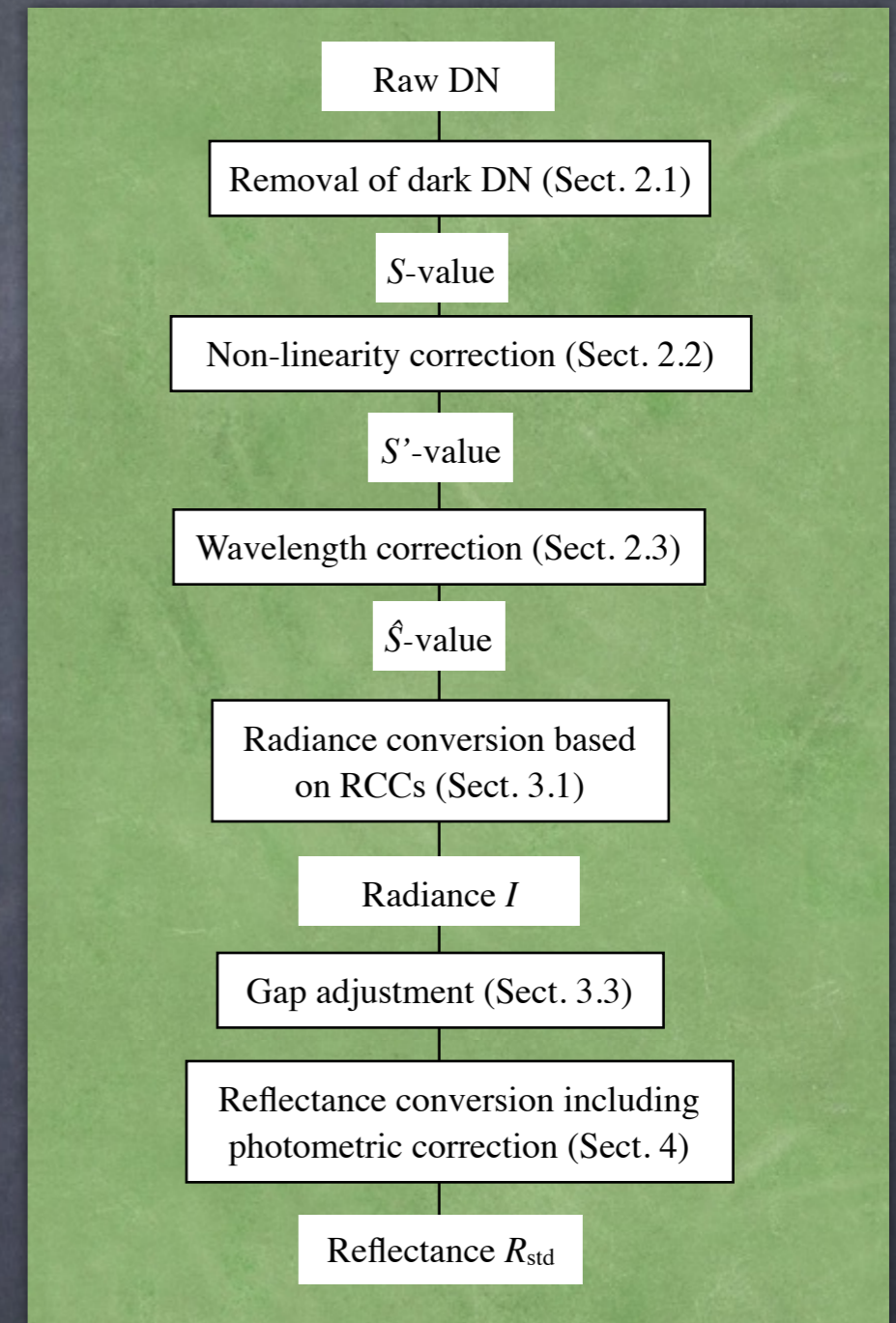
(2)  $S$ 値に対して非線形補正、波長ズレ補正等の校正処理

→  $\hat{S}$ 値

$$I(n) = \frac{\hat{S}(n)}{C(n)}$$

$I(n)$ : 輝度値

$C(n)$ : 輝度値変換係数(RCC)

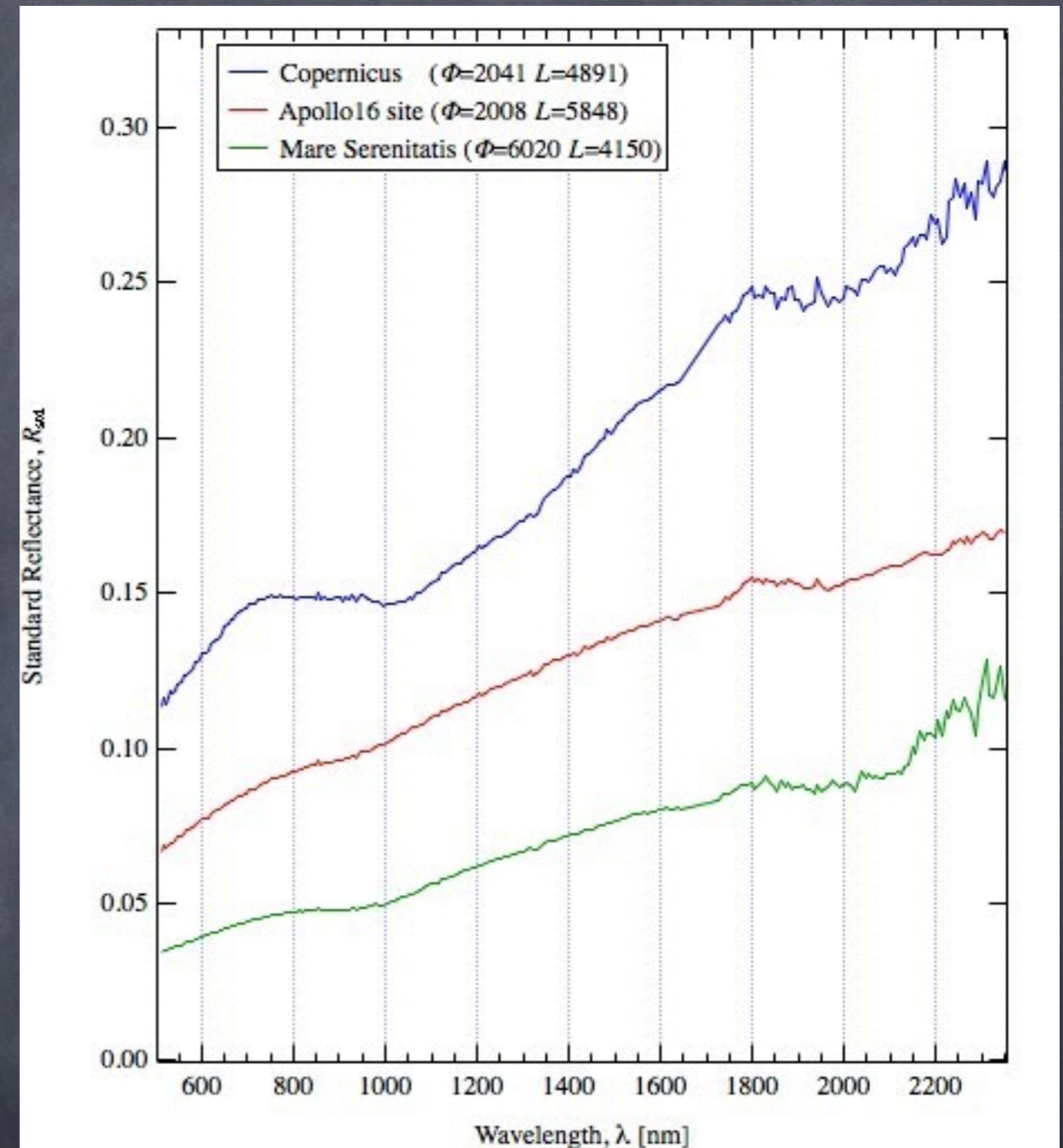
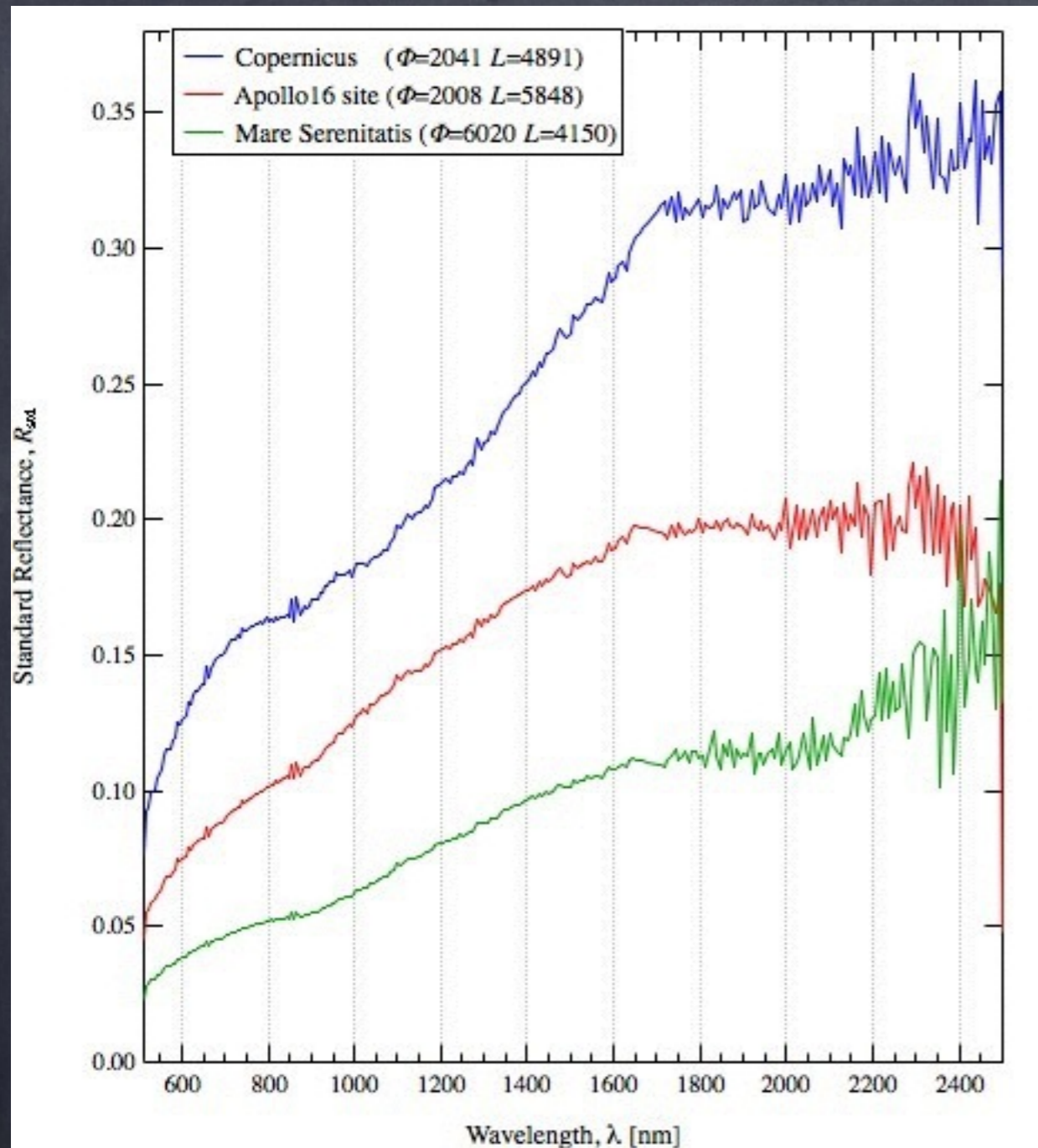


SPプロダクト生成の流れ

Yamamoto et al. (2011 & 2014 TGRS)

このRCCに含まれる固有に  
フィーチャー

## 4.1 反射率補正係数とは



SPプロダクト(L2C)

反射率補正処理後

### (応用演習) 反射率補正係数の作成

(1) A16 soilデータとSPによるA16着陸地点観測データを使って、補正係数を決定

(2) SP in Cに補正係数を組み込む作業

## 4.2 反射率補正係数の作成

### 作業1 : SPによるA16着陸地点観測データの取得

- SPによるA16着陸地点観測(ASO)に対応するデータを調べる
- JAXAサイトから必要データ (L2B1) のダウンロード
- SPINC\_simplereadL2Bを使ってファイルを出力

使用しない素子の補正係数は今は除去しないで係数決定。  
→係数を実際に使用する段階で除去する (後の組み込み処理が楽になる)。

Yamamoto et al.,  
2011, TGRS, P.4672の

表4にASOデータ

- 表4はL2B1データに対する情報
- どの周回番号、どの側線データを使うかは、任意

4672

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 49, NO. 11, NOVEMBER 2011

TABLE IV  
CONDITIONS OF THE APOLLO 16 LANDING SITE OBSERVATIONS

	1st	2nd	3rd	4th
Revolution, $\Phi$	503	2008	4175	6375
Date	November 19, 2007	March 21, 2008	September 15, 2008	March 12, 2009
Line number range, $L$	4545 to 4551	5844 to 5850	4961 to 4967	5804 to 5810
†Solar Incident Angle, $i$	44.73°	14.84°	22.19°	25.26°
†Solar Emission Angle, $e$	9.84°	13.32°	2.75°	4.57°
†Solar Phase Angle, $g$	35.19°	26.16°	24.78°	29.83°

†The average value of 7 lines

L2B1に対するLine番号

Fig. 19 shows  $R_{std}$  for the three reference sites. It is shown

## 4.2 反射率補正係数の作成

### 作業 2 : A16着陸地点のSoilデータ

本来必要な作業

- 1) RELABからの取得
- 2) SPの波長に対するリサンプリング

- ・今日は予め用意されているSPの波長にあわせたデータを使用

Resampled\_SOILDATA\_62231.txt

やってみよう！

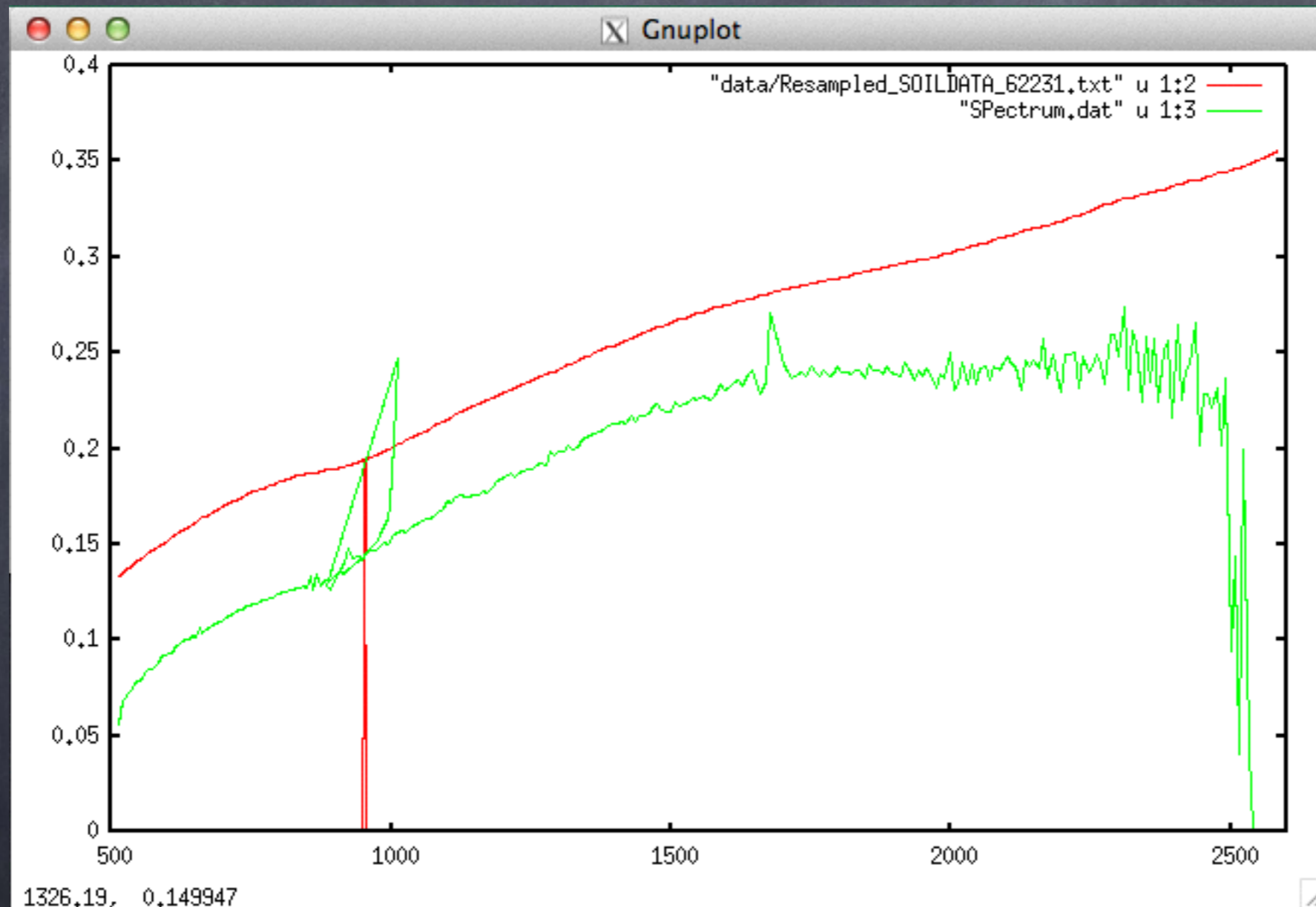
(a) Resampled\_SOILDATA\_62231.txtのグラフ化

(b) 作業1で得られたデータの比較



## 4.2 反射率補正係数の作成

```
gnuplot> p "data/Resampled_SOILDATA_62231.txt" u 1:2 w l, "Spectrum.dat" u 1:3 w l
```



## 4.2 反射率補正係数の作成

### 作業3：補正係数の決定

(1) 補正係数 = soilデータ ÷ SPデータ

(2) 752.8 nm (41素子目) で規格化

(3) 結果を"Correction\_Factor.txt"という名前で保存。

・以上の計算をプログラムまたは、表計算ソフト等を使って求めてみよう。

## 4.2 反射率補正係数の作成

サンプルプログラム

Calculating\_Correction\_Factor.c

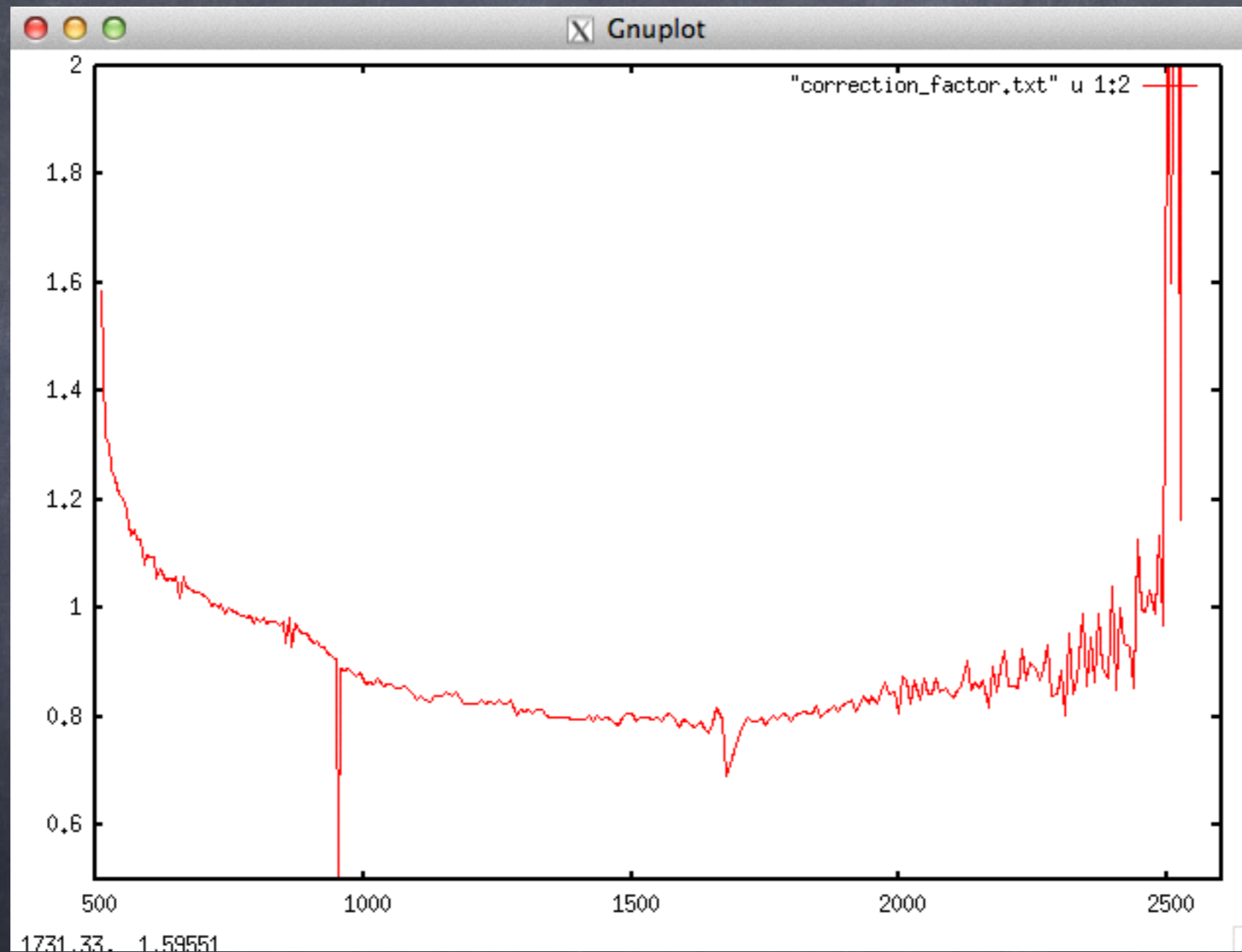
マクロ: ↕

```
1  /***** ↓
2  Correnction Factor Calculation ↓
3  *****/ ↓
4  #include <stdio.h> ↓
5  #include <stdlib.h> ↓
6  ↓
7  int main () { ↓
8  __int px, scale_pixel_numer = 39, BAND_NUM = 296; ↓
9  __double scale_value, wavelength_data[ BAND_NUM ], soil_data[ BAND_NUM ], reflectance_data[ BAND_NUM ], correction_factor[ BAND_NUM ]; ↓
10 __FILE *OUTPUT_FILE_NAME, *INPUT_FILE_NAME_A, *INPUT_FILE_NAME_B; ↓
11 __//File Open ↓
12 __INPUT_FILE_NAME_A = fopen("Resampled_SOILDATA_62231.txt", "r"); ↓
13 __INPUT_FILE_NAME_B = fopen("Spectrum.dat", "r"); ↓
14 __OUTPUT_FILE_NAME = fopen("correction_factor.txt", "w"); ↓
15 __//Reading & Calculation ↓
16 __for(px=0; px<BAND_NUM; px++){ ↓
17 ____fscanf(INPUT_FILE_NAME_A, "%lf %lf", &wavelength_data[ px ], &soil_data[ px ]); ↓
18 ____fscanf(INPUT_FILE_NAME_B, "%*lf %*lf %lf %*d %*d %*d", &reflectance_data[ px ]); ↓
19 ____(correction_factor+ px) = *(soil_data+px) / *(reflectance_data + px); ↓
20 __} ↓
21 __//scaling value at 752.8 nm ↓
22 __scale_value = *(correction_factor + scale_pixel_numer); ↓
23 __//Output ↓
24 __for(px=0; px<BAND_NUM; px++) fprintf(OUTPUT_FILE_NAME, "%.1f %f\n", *(wavelength_data + px), *(correction_factor + px)/scale_value); ↓
25 __//File close ↓
26 __fclose( INPUT_FILE_NAME_A ); ↓
27 __fclose( INPUT_FILE_NAME_B ); ↓
28 __fclose( OUTPUT_FILE_NAME ); ↓
29 __return 0; ↓
30 } ↓
31 //EOF ↓
```

データをおいて  
いる場所に注意

## 4.2 反射率補正係数の作成

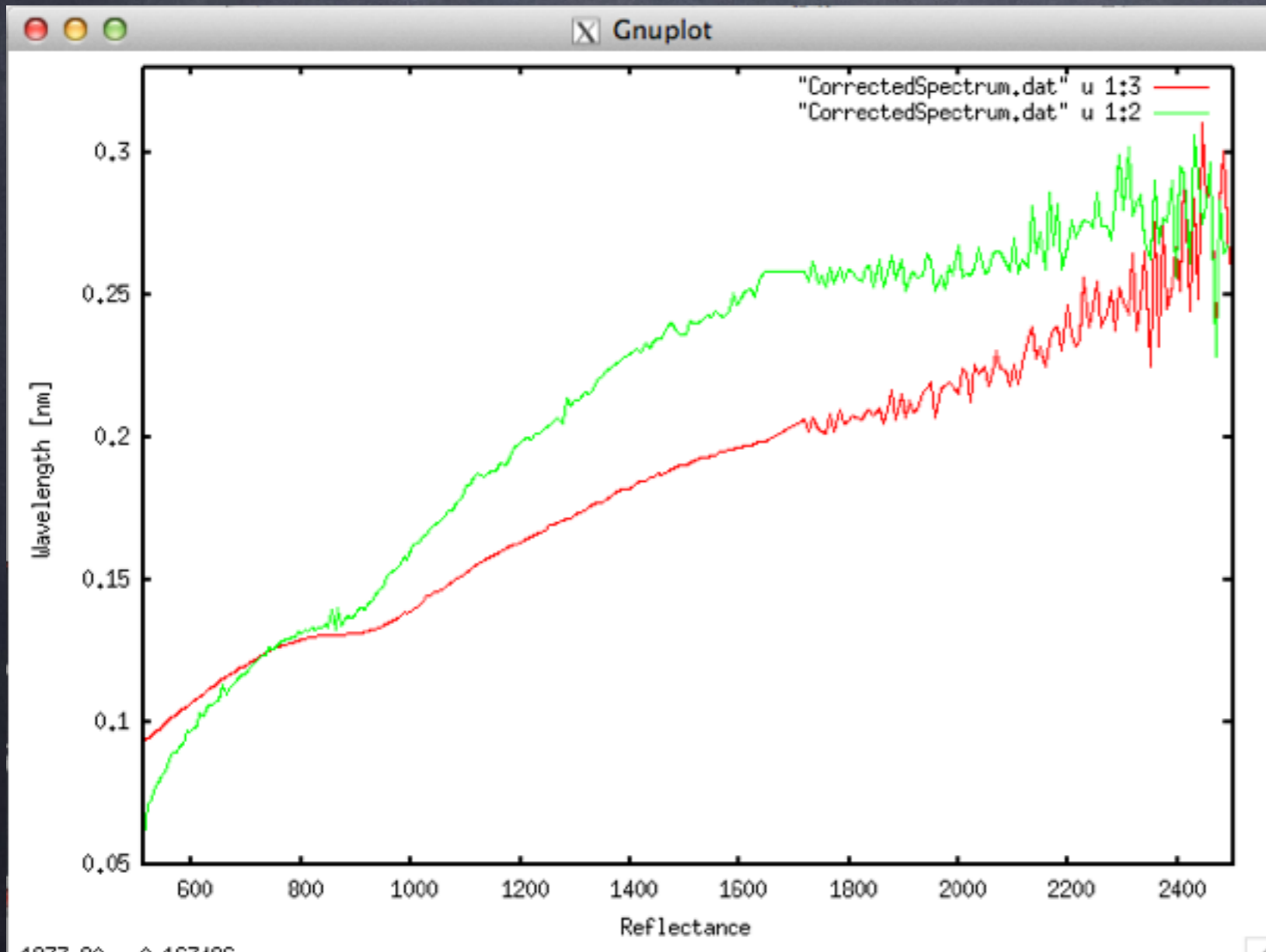
### Correction Factor.txtの例



## 4.3 補正係数のSP in Cへの組み込み

### 作業4：補正係数のMAIN\_simple\_reading\_L2B.cへの組み込み

- ・ Correction\_Factor.txtを読み込み、SPプロダクトの反射率データに掛け算するプログラム



緑に補正係数を  
かけた結果が赤