

修 士 学 位 論 文

Lomb-Scargle法を用いた6.7GHzメタノールレーザー 強度変動周期の導出

2015年度
(平成27年度)

茨城大学大学院理工学研究科

理学専攻

14NM166T

安井靖堯

概要

本研究は、茨城観測局日立 32m 電波望遠鏡 (以下、日立アンテナ) でモニター観測している 461 天体の 6.7GHz メタノールメーザーから、大質量星形成過程の解明の手がかりになると期待されている、周期的な強度変動を示す天体を定量的に選出する解析手法の確立を目指す。

大質量星形成過程の解明の指標とされている 6.7GHz メタノールメーザーの中には、強度変動を示すものが数多く報告されている (e.g., Goedhart et al. 2004)。中でも、周期的な強度変動を示す天体が観測されているのが特徴的である (e.g., Goedhart et al. 2014; Szymczak et al. 2015)。しかし、周期的な強度変動を発生させるメカニズムは解明されていない。現在、周期強度変動メカニズムを説明するのに最も有力とされる理論は、Inayoshi et al. (2013) による大質量原始星の脈動不安定から予想される理論である。そこで、日立アンテナでは、この理論の観測的検証を目的の一つとして長期間にわたるモニター観測を実施している。しかし、これまで日立アンテナ・モニター観測によって得られたデータから強度変動の周期性の有無や強度変動周期の導出を行う方法は、目視などによる定量的ではない方法しかなかった。そこで、本研究は、日立アンテナ・モニター観測によって得られたデータから周期的な強度変動を示す 6.7GHz メタノールメーザーを定量的に選出する手法の確立を目指した。一般的に用いられる周期解析法は、データとデータの間隔、すなわち観測日間隔が等間隔の場合に適用可能なものが多い。そのため、観測装置の運用上の都合や天候の影響などにより、必ずしも等間隔に観測を行えないことが多い天文観測のデータは解析することができない。そこで、本研究では、観測日間隔が不等間隔であっても周期解析が可能である Lomb-Scargle 法 (Lomb 1976, Scargle 1982, etc.) を用いた。Lomb-Scargle 法では、任意の観測日間隔で得られた時系列データのパワースペクトルを推定する。すなわち、ピリオドグラムを導出し、その最大値を示す周波数の逆数を周期として読み取る。本研究では、同時に、Lomb-Scargle ピリオドグラムによって求まる最大値の有意性を計るため、統計学の仮説検定を用いて、帰無仮説「観測値は正規分布する雑音のみである」の誤警報確率を求める。

解析の実行には、Numerical Recipes in C (Press et al. 1988) を参考に、C 言語プログラムを作成した。解析可能なデータが存在する 443 天体に対して Lomb-Scargle 法を実行し、選出条件を課すことによって、周期的な強度変動を示す天体かどうかを分別した。先行研究では、既知の周期強度変動天体は 17 天体であると報告されている。その内、日立アンテナ・モニター観測の対象である 12 天体から 10 天体が選出条件をすべて満たした。ただし、その内 1 天体は先行研究とは大きく異なる周期が導出された。また、既知の周期強度変動天体とは別に、新たに 11 天体を周期強度変動天体として選出した。本章では、これら 12 天体を簡単に紹介していく。

目次

1	イントロダクション	1
1.1	大質量星	1
1.2	大質量星形成領域とメーザー	2
1.3	研究目的	5
2	Lomb-Scargle 法	6
2.1	本研究で扱うデータ	6
2.2	ピリオドグラム	7
2.3	Lomb-Scargle ピリオドグラム	8
2.4	Lomb-Scargle ピリオドグラムによって与えられる独立周波数の数	10
2.5	Lomb-Scargle ピリオドグラムの統計分布	11
3	解析プログラム	16
3.1	日立アンテナ・モニター観測	16
3.2	データの前処理	17
3.3	オーバーサンプリング	18
4	結果・考察	20
4.1	選出条件	20
4.2	新検出の周期強度変動天体	21
4.3	選出除外天体	44
4.4	考察	46
5	まとめ	49
付録 A	メーザー	50
付録 B	定常過程とエルゴート過程	50
付録 C	Lomb-Scargle ピリオドグラムの時不変性	51
付録 D	プログラム lsperiodogram	52
D.1	プログラム lsperiodogram のソースコード	53
D.2	プログラムの実行方法	62
D.3	関数の説明	63
D.4	プログラムの内容説明	63

1 イントロダクション

1.1 大質量星

星形成過程や星形成領域の解明は、宇宙や銀河の成り立ちを研究することへつながるため、天文学において重要なテーマの一つとされている。中でも、大質量星 (太陽質量の 8 倍以上を持つ星; $\geq 8M_{\odot}$) の形成過程は完全に解明されていない。その上、その存在の影響力は大きいと言える。具体的には、

- 大質量星の光度は明るいため、銀河全体の明るさに大きく寄与すること。
- 大質量星が最終的に引き起こす超新星爆発による衝撃波は、新たな星形成の引き金に成り得ること。同時に、核融合反応によって生成された重元素合成を星間空間へ供給すること。
- 強い紫外線源となり、周囲のガス (水素) を電離・加熱させ、電離領域 (HII 領域) を形成すること。

これらのように、大質量星は、宇宙や銀河スケールで大きな役割を果たす存在である。しかし、次の困難から、大質量星の形成過程は未だ解明されていない。

- 質量降着によって星が進化するタイムスケールと重力収縮するタイムスケールを比べると、後者の方が短くなる。そのため、大質量星は主系列星になった後でも、質量降着を続けることになる。
- 大質量星の質量になると、輻射圧によって質量降着が止められる可能性が指摘されている。

したがって、大質量星の質量降着による形成過程は完全には理解されていない。また、次の理由から、大質量星の直接観測は難しいと言える。

- その寿命が $10^6 - 10^7$ 年であること。太陽 (寿命は約 10^{10} 年) を一般的な星とした場合、比較すると短いことがわかる。
- 短寿命なことに加え、形成頻度も低いため絶対数が少ないこと。したがって、形成初期のものは、さらに少ないと言える。
- 形成初期の原始星段階 ($\sim 10^4 - 10^5$ 年) は濃い分子雲に埋もれているため、可視光を放射しても吸収される。
- ほとんどが地球から 1kpc 以遠に存在している。形成領域の大きさは $\sim 10 - 1000\text{AU}$ 、原始星に至っては $< 0.1\text{AU}$ であるため、1kpc の距離にある場合、1 秒角よりも高分解能な観測が必要になる。

1.2 大質量星形成領域とメーザー

大質量星形成領域，特に原始星の近傍から H_2O や OH ， CH_3OH などの分子のメーザー放射 (付録 A) が観測されている。メーザーは，放射強度が強く，放射サイズが小さい ($\sim 1\text{AU}$) ため，VLBI などの高い空間分解能観測 (10^{-3}arcsec) により，大質量星形成領域の空間分布を観測可能である。また，メーザーは，ドップラー効果による視線方向の速度と，固有運動による接線方向の速度の情報が観測可能であるため，天体の運動を 3 次元情報として解明するのに有益な存在である。視線方向の速度に関しては，単一鏡で観測可能であり，視線速度に対する (エネルギー) フラックス密度の情報を入手可能である。(エネルギー) フラックス密度とは，単位時間・単位面積・単位周波数幅当たりの電磁波のエネルギーである。単位には， $[\text{Jy}]$ (ジャンスキー) が用いられる。 $1[\text{Jy}] = 10^{-26}[\text{W m}^{-2} \text{Hz}^{-1}]$ である。

1.2.1 6.7GHz メタノールメーザー

メタノールメーザーのうち，静止周波数が $6668.5192 \pm 0.0008\text{MHz}$ (Breckenridge & Kukolich 1995) であるものを 6.7GHz メタノールメーザー^{*1} という。6.7GHz メタノールメーザーは，大質量星形成領域にのみ付随しており，多くは原始星段階に検出されている (e.g., Walsh et al. 1998)。したがって，6.7GHz メタノールメーザーは，可視光では直接観測が困難な大質量星形成領域を間接的に観測可能にする指標となる。6.7GHz メタノールメーザーの特徴として，全メーザー種の中で H_2O メーザーについて 2 番目に明るいこと，放射の寿命が約 4 年以上であることが挙げられる。6.7GHz メタノールメーザーの典型的な輝度温度 ($> 10^{12}\text{K}$) を持つには，中心星によって暖められたダスト ($\sim 150\text{K}$) からの赤外線によって励起される必要がある (Cragg et al. 2005)。また，6.7GHz メタノールメーザーの中には強度変動を示す天体も観測されている (e.g., Goedhart et al. 2004)。

1.2.2 周期的な強度変動を示す 6.7GHz メタノールメーザー

強度変動を示す 6.7GHz メタノールメーザーのうち，周期的な強度変動を示すものが発見されている。例えば，Goedhart et al. (2004) による，54 天体を対象にした 4 年間のモニター観測では，7 天体の強度変動に厳密な周期性があると結論付けている。6.7GHz メタノールメーザーの周期的な強度変動は，中心星の光度変化の影響を直接受けていると考えられるため，その詳細を調べることは大質量星形成領域の詳細を知る手がかりになる。現在，およそ 1000 個以上の 6.7GHz メタノールメーザー源が発見されており，その内，周期的な強度変動を示す天体は 17 天体発見されている (表 1: Goedhart et al. 2004, 2009, 2014; Szymczak et al. 2011, 2015; Maswanganye et al. 2015; Araya et al. 2010; Fujisawa et al. 2014)。しかし，周期的な強度変動が起こる原因は解明されておらず，未だ大質量星形成領域の詳細を知るまでに至っていない。

^{*1} もしくは，励起状態 $(J, K) = (5, 1)$ から低励起状態 $(J, K) = (6, 0)$ への A^+ 軸対称に遷移 $5_1 \rightarrow 6_0 A^+$ と表す。

表 1 既知の周期的な強度変動を示す 6.7GHz メタノールメーザー

天体名	周期 [day]	参考文献
G 009.62+00.19	243	Goedhart et al. 2014
G 012.68-00.18	307	Goedhart et al. 2004
G 012.88+00.48	29.5	Goedhart et al. 2009
G 022.36+00.06	179.3	Szymczak et al. 2015, 2011
G 025.41+00.10	244.7	Szymczak et al. 2015
G 037.55+00.20	237	Araya et al. 2010
G 045.47+00.13	195.6	Szymczak et al. 2015
G 073.06+01.80	159	Szymczak et al. 2015
G 075.76+00.34	119.9	Szymczak et al. 2015
G 188.94+00.88	416	Goedhart et al. 2004
G 196.45+01.67	668	Goedhart et al. 2004
G 328.23-00.54	220.5	Goedhart et al. 2014
G 331.13-00.24	509	Goedhart et al. 2014
G 338.93-00.06	132.8	Goedhart et al. 2014
G 339.62-00.12	200.3	Goedhart et al. 2014
G 358.46-00.39	220	Maswanganye et al. 2015
IRS22198+6336	34.6	Fujisawa et al. 2014

1.2.3 周期的な強度変動を示す 6.7GHz メタノールメーザーの発生機構候補

6.7GHz メタノールメーザーの周期的な強度変動の発生機構は、まだ解明されていない。これまでいくつかの議論 (e.g., van der Walt 2011; Parfenov & Sobolev 2014; Goedhart et al. 2008) がなされているが、現在、その発生機構を説明する最も有力な理論は、Inayoshi et al.(2013) によって提唱される大質量原始星の脈動不安定によるものである。現在、提唱されている大質量星形成のシナリオによると、大質量星形成には質量降着率 $\dot{M}_* \geq 10^{-3} M_\odot \text{yr}^{-1}$ が要求される。大質量星形成シナリオでは、この質量降着率のもとで、原始星の半径は質量の増加とともに大きくなる。しかし、 $M_* \geq 10 M_\odot$ になると、原始星は放射によってエネルギーを放出し、収縮を始める (Kelvin-Helmholtz 収縮)。その後、 $M_* \approx 40 M_\odot$ になると、原始星の半径は再び大きくなり始める。原始星の半径の進化が非線形になる領域 (不安定帯) にあるとき、衝撃波によって脈動のエネルギーが放出される。その放出されたエネルギーの一部が周期的なアウトフローの運動エネルギーになる。Inayoshi et al.(2013) によると、原始星の半径が最大のときのみ、原始星は脈動不安定になる。図 1^{*2} は、様々な質量降着率ごとの質量と原始星半径の脈動の様子を示した図である (Inayoshi et al. 2013; Figure 1)。Inayoshi et al.(2013) では、原始星の脈動周期 P_* と光度 L_* の

^{*2} Young Stellar Object(YSO) 付随のものと HII 領域付随のもので分けている。また、変動傾向を間欠的/連続的で分けている。2 天体を除く理由は、それぞれ対応する IRAS 源が見つからなかったためと、中質量星に付随している可能性があるからである。

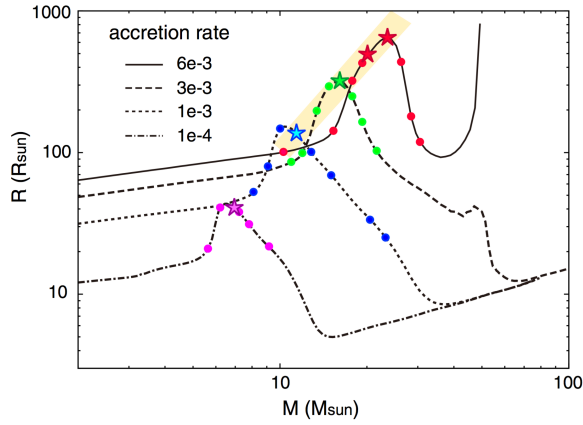


図 1 様々な質量降着率 (10^{-4} , 10^{-3} , 3×10^{-3} , $6 \times 10^{-3} M_{\odot} \text{yr}^{-1}$) の質量と原始星半径の脈動の様子 (Inayoshi et al. 2013; Figure 1)。●と☆印は、それぞれ脈動安定と脈動不安定を表しており、影の部分は不安定帯を指す。

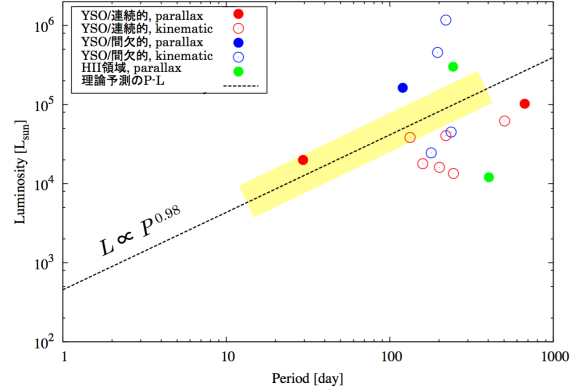


図 2 原始星の脈動周期と光度の関係 (杉山孝一郎作成 (茨城大学); Inayoshi et al. 2013; Figure 2 改)。影は、図 1 と同じく不安定帯を表している。破線は式 1 を示したものである。また、表 1 に示した 6.7GHz メタノールメーザーの内、G 012.68+00.18 と IRAS22198+6336 を除く天体を距離決定法別に●と○で示してある。

関係が、

$$\log \left(\frac{L_*}{L_{\odot}} \right) = 4.62 + 0.98 \log \left(\frac{P_*}{100[\text{days}]} \right) \quad (1)$$

であると予言している。また、周期的な強度変動を示す 6.7GHz メタノールメーザーの変動周期と光度も、この関係にある可能性を提唱している。したがって、周期的な強度変動を示す 6.7GHz メタノールメーザーは、大質量原始星時代に生じると予言されている脈動不安定によって説明可能であると提唱している。ただし、脈動自体は κ 機構と呼ばれるメカニズムに起因しており、原始星半径がわずかに変動する。同時に、有効温度もわずかに変動するため、メタノール分子が存在するダストの温度変化をもたらし、メーザーの励起環境を変動させることになる。結果として 6.7GHz メタノールメーザーの強度も周期的に変動する可能性が考えられる。図 2 は、原始星の脈動周期と光度の関係、周期的な強度変動を示す 6.7GHz メタノールメーザーの変動周期と光度を示した図である (杉山孝一郎作成 (茨城大学); Inayoshi et al. 2013; Figure 2 改)。この理論が実証されれば、式 (1) で与えられる関係から、周期的な強度変動を示す 6.7GHz メタノールメーザーを観測することによって、大質量原始星の物理量 (質量、半径など) を推定できることへもつながる。

1.3 研究目的

本研究の目的は、茨城観測局日立 32m 電波望遠鏡 (以下、日立アンテナ) を用いたモニター観測 (詳細は後述 §3.1) によって得られた 6.7GHz メタノールメーザーのデータから、周期的な強度変動を示す天体を定量的に選出する方法の確立である。日立アンテナは、§1.2.3 で説明した Inayoshi et al.(2013) によって提唱される周期的な強度変動を示す 6.7GHz メタノールメーザーの発生機構の観測的実証を主な目的の一つとしてモニター観測を実施している。すなわち、日立アンテナ・モニター観測は、既知の周期強度変動天体 (表 1) に加わる新たな周期的な強度変動を示す 6.7GHz メタノールメーザーを見つけることを目的の一つとしている。周期的な強度変動を示す 6.7GHz メタノールメーザーを多く見つけることにより、式 (1) の関係を検証する。ただし、これまで、日立アンテナ・モニター観測によって得られたデータから周期性の有無や周期の導出を行う方法は、目視によるなど定量的でない方法しかなかった。そこで、本研究では、日立アンテナ・モニター観測によって得られたデータから、周期的な強度変動を示す 6.7GHz メタノールメーザーを定量的に選出する手法を確立する。定量的な方法を用いることによって、導出される周期自体の信頼度を与えられるため、今後、式 (1) の関係などの検証に用いることができる。

2 Lomb-Scargle 法

2.1 本研究で扱うデータ

本研究では、モニター観測 (詳細は後述 §3.1) によって得られた各日の 6.7GHz メタノールメーザーの強度変動を時系列データとして扱う。ここでの強度とは、正確にはフラックス密度 [Jy] である。以下では、この時系列データのことを観測値や観測データと呼ぶことがある。時系列データを解析することを時系列解析もしくは信号処理という。一般的な信号処理で扱うデータは、信号 (signal) と雑音 (noise) に分けることができる。本研究では、周期的な強度変動を示す天体の強度を「信号」と称し、非周期的な強度変動や変動を示さない強度、観測装置による熱雑音など、周期的な強度変動成分以外を総じて「雑音」と称す。簡単のため、雑音 $R(t_i)$ は、独立 ($R(t_i) \neq R(t_j)$, $i \neq j$) であり、正規分布 $N(0, \sigma_0^2)$ ^{*3} に従うと仮定する。このとき、特に非周期的な強度変動や変動を示さない強度の分布は、正規分布では近似しにくいことは念頭に置いておく必要がある。また、観測値は、すべての観測期間で同じ確からしさであると仮定する。

時系列解析では、データに長期間にわたる増加や減少が含まれることがある。この性質のことをトレンドといい、時系列データにトレンド成分が含まれるという。言い換えると、トレンドとは、時系列解析によって求めたい周波数よりも非常に低い周波数成分にあたる。以上を踏まえ、本研究で扱うデータは、信号と雑音とトレンドの和で表されると仮定する。

信号処理や時系列解析では、連続信号から一定の間隔でサンプリングした離散信号を扱ったり、毎日の売上を時系列データとして扱ったりと、データ点とデータ点の間は等間隔であることが多い。しかし、天文観測では巨大な観測装置の運用上の都合や天候の影響などにより、必ずしも等間隔の時間に観測を行えないことが多い。このような理由によって、本研究で扱うデータは、観測日間隔が任意の間隔 (等間隔と不等間隔) で得られたデータである。そのため、解析方法には、等間隔で得られたデータを前提にした理論や手法を適用することはできない。

式で表すと、本研究で扱うデータは、 N_0 個の任意の観測間隔で得られた離散時系列データ $X(t_i)$, $i = 1, 2, \dots, N_0$ であり、周期関数 $X_S(t_i)$ と雑音 $R(t_i)$ とトレンド成分 m_t の和、

$$X(t_i) = X_S(t_i) + R(t_i) + m_t \quad (2)$$

となる。ただし、簡単のため、この章の議論で扱うデータに限っては、トレンド成分を含まない、

$$X(t_i) = X_S(t_i) + R(t_i) \quad (3)$$

を用いる。

^{*3} $N(\mu, \sigma^2)$ は平均 μ , 分散 σ^2 の正規分布 (ガウス分布) を表す。このとき、正規分布の確率密度関数は、

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(x - \mu)^2}{2\sigma^2} \right]$$

で与えられる。

2.2 ピリオドグラム

式 (3) で与えられるデータの離散フーリエ変換 (DFT) は,

$$\begin{aligned} DFT(\omega_n) &= \sum_{j=1}^{N_0} X(t_j) \exp(-i\omega_n t_j) \\ &= \sum_{j=1}^{N_0} X(t_j) \exp\left(-i \frac{2\pi n}{\Delta t N_0} t_j\right) \end{aligned}$$

で与えられる。ここで、 $n = 1, 2, \dots, N_0$ である。これを用いて、パワースペクトルの推定に当たるピリオドグラムは,

$$P(\omega_n) = \frac{1}{N_0} |DFT(\omega_n)|^2 \quad (4)$$

で与えられる。 N_0 個の離散時系列データに含まれる周期関数 $X_S(t_i)$ の周波数が ω_{n_0} である場合、ピリオドグラムは $P(\omega_{n_0})$ で大きな値を取り得る。このように時系列データに含まれる周期をピリオドグラムから捉えることができる。なお、強度が一定である時系列データのピリオドグラムは、 $\omega = 0$ に最大値を持つことが期待される。特に、この $\omega = 0$ のパワーを直流成分 (DC 成分) という。

観測間隔 Δt が等間隔の場合、ピリオドグラムによって導出される正味の独立周波数の数 N は、 $P(\omega) = P^*(\omega)$ となることを踏まえると,

$$N = \frac{N_0}{2} \quad (5)$$

である。特に、導出可能な最大周波数 $\omega_c = \pi/\Delta t$ をナイキスト周波数という。表 2 に、等間隔の時系列データに対する各物理量をまとめた。

次に、データ数 N_0 を変化させずに、独立周波数の周波数分解能をより細かくしたい場合を考える。元の時系列に,

$$X(t_j) = 0, j = N_0 + 1, \dots, M$$

表 2 等間隔の時系列データの各物理量

物理量	式
データ数	N_0
ナイキスト周波数 ω_c	$2\pi \frac{N_0}{2T}$
離散フーリエ変換 $DFT(\omega_n)$	$\sum_{j=1}^{N_0} X(t_j) \exp(-i\omega_n t_j)$
周波数分解能	$\frac{2\pi}{\Delta t N_0}$
独立周波数の数 N	$\frac{N_0}{2}$

表 3 オーバーサンプリング時の等間隔の時系列データの各物理量

物理量	式
データ数	mN_0
ナイキスト周波数 ω'_c	$2\pi \frac{mN_0}{2T}$
離散フーリエ変換 $DFT'(\omega_n)$	$\sum_{j=1}^{mN_0} X(t_j) \exp(-i\omega_n t_j)$
周波数分解能	$\frac{2\pi}{\Delta t m N_0}$
独立周波数の数 N'	$\frac{mN_0}{2}$

となる $M - N_0$ 個の 0 点を挿入し、元の時系列の時間軸の拡大を計る。このとき、離散フーリエ変換は、

$$DFT(\omega_n) = \sum_{j=1}^M X(t_j) \exp(-i \frac{2\pi n}{\Delta t M} t_j)$$

で与えられる。したがって、周波数分解能は $(2\pi)/(\Delta t M)$ となるため、0 点の挿入前より細かくすることができる。このことは、データ数を M 個にしたことに相当するので、各物理量は表 3 のように与えられる。 M は $M \geq N_0$ であれば自由に選ぶことができるが、しばしば N_0 の整数倍で選ぶことが多い。このとき、 $M = mN_0$ (m は 2 以上の整数) とすることを、 m 倍にオーバーサンプリングするという。特に、離散時間のデータに対するオーバーサンプリングは、0 点を加えることから補間とも言われる。

2.3 Lomb-Scargle ピリオドグラム

等間隔に観測されたデータ $X(t_i)$ が正規分布 $N(0, \sigma_0^2)$ に従う雑音 $R(t_i)$ のみの場合、ピリオドグラム (式 (4)) の確率分布は指数分布^{*4} となる。このとき、確率分布関数 $f(x)$ は、

$$f(x) = \frac{1}{\sigma_0^2} \exp\left(-\frac{x}{\sigma_0^2}\right)$$

で表される。一方、不等間隔に観測されたデータ $X(t_i) = R(t_i)$ のピリオドグラムの確率分布は、指数分布になるとは限らない。一般に、確率分布の分からない、もしくは推定できない時系列デー

^{*4} もしくは、ガンマ分布 $\Gamma(1, \sigma)$ という。また、 $N(0, 1)$ に従う雑音のピリオドグラム (式 (4)) の場合、自由度 2 の χ^2 分布である。ここで、標準正規分布 $N(0, 1)$ に従う k 個の独立変数 $Z_1, Z_2, \dots, Z_k$ の 2 乗和

$$\chi_k^2 = Z_1^2 + Z_2^2 + \dots + Z_k^2$$

の確率分布を自由度 k の χ^2 分布という。

タを扱うことは非常に困難である。しかし、ピリオドグラム (式 (4)) に,

$$P_X(\omega) = \frac{1}{2} \left\{ \frac{\left[\sum_{j=1}^{N_0} X(t_j) \cos \omega(t_j - \tau) \right]^2}{\sum_{j=1}^{N_0} \cos^2 \omega(t_j - \tau)} + \frac{\left[\sum_{j=1}^{N_0} X(t_j) \sin \omega(t_j - \tau) \right]^2}{\sum_{j=1}^{N_0} \sin^2 \omega(t_j - \tau)} \right\} \quad (6)$$

ここで τ は,

$$\tan(2\omega\tau) = \frac{\sum_{j=1}^{N_0} \sin 2\omega t_j}{\sum_{j=1}^{N_0} \cos 2\omega t_j}, \quad (7)$$

で与えられる修正を加えると, 等間隔の観測データ $X(t_i) = R(t_i)$ のピリオドグラムと同様に指数分布となる。このとき, 分散 σ_0^2 が 1 でない観測データ $X(t_i) = R(t_i)$ の場合, 観測データ $X(t_i) = R(t_i)$ の (不偏) 分散,

$$\sigma_0^2 = \frac{1}{N_0 - 1} \sum_{i=1}^{N_0} (X(t_i) - \bar{X})^2, \quad (8)$$

ここで, $\bar{X}$ は観測データ $X(t_i) = R(t_i)$ の平均値,

$$\bar{X} = \frac{1}{N_0} \sum_{i=1}^{N_0} X(t_i), \quad (9)$$

を用いて,

$$P_N(\omega) = \frac{P_X(\omega)}{\sigma_0^2} \quad (10)$$

のように正規化することにより, 確率密度関数が $f(x) = \exp(-x)$ の指数分布を持たせることができる。式 (6) は, はじめ Lomb(1976) によって提唱された式であり, 後に Scargle(1982) によって詳細に評価されたものである。式 (6) を Lomb-Scargle ピリオドグラムと呼ぶことにし, 式 (4) のピリオドグラムとは区別する。式 (6) を正規化した式 (10) は, Horne & Baliunas(1986) によって提唱された正規化の方法であり, 式 (10) を正規化 Lomb-Scargle ピリオドグラムと呼ぶことにする。また, 式 (6), 式 (10) を用いた計算を Lomb-Scargle 法と呼ぶことにする。

次の 2 つの理由から, Lomb-Scargle ピリオドグラムは, 観測日間隔が不等間隔であるデータであっても, 等間隔で得られたデータと等価のピリオドグラムを求めることができる。第一に, 観測データが定常過程 (付録 B) である場合, τ を与えることにより, 式 (6) は時間シフトに対して不変となるためである (付録 C)。簡単に言えば, 式 (6) はデータを時間間隔ごとに取り扱うのではなく, データ点ごとに取り扱うことができる。

第二に, 式 (6) の形は, 観測データ $X(t_i)$ を,

$$X_f(t_i) = A \cos(\omega t) + B \sin(\omega t),$$

にフィットさせるために,

$$E(\omega) = \sum_{j=1}^{N_0} [X(t_j) - X_f(t_j)]^2$$

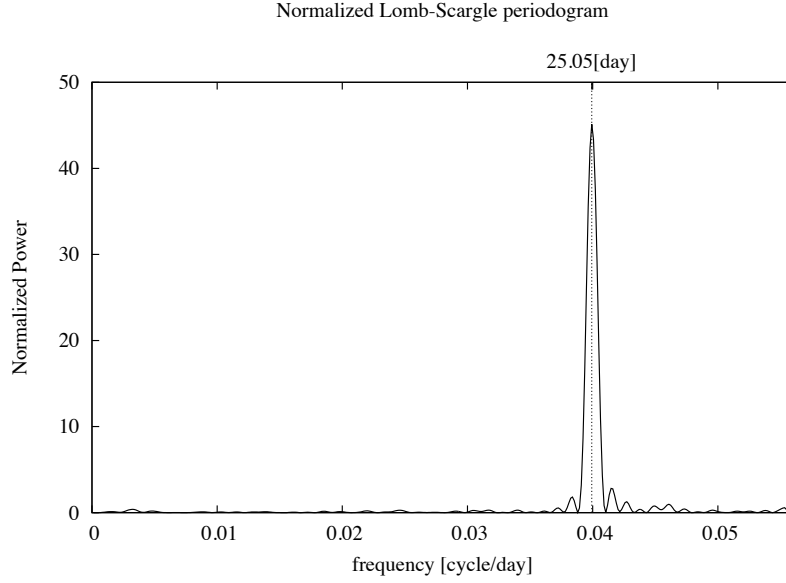


図3 周期 25 日 ($\omega = 0.04[\text{rad/day}]$) の正弦波と正規分布 $N(0, 1)$ に従う雑音の和で与えられる不等間隔の擬似データの正規化 Lomb-Scargle ピリオドグラムのスペクトル

を最小にする方法と等価であるためである。ここで、 A 、 B は定数である。すなわち、観測データを最小二乗法を用いて正弦波にフィットさせることと同じである。フィットさせた関数 $X_f(t_i)$ の周波数を読み取ることで、周期を求めることができる。この導出方法は、Lomb(1976) によって与えられたものである。具体的に、どのようにフィットさせるかについては Lomb(1976) を参照していただきたい。

図 3 に、参考例として周期 25 日 ($\omega = 0.04[\text{cycle/day}]$) の正弦波と正規分布 $N(0, 1)$ に従う雑音の和で与えられる不等間隔の擬似データの正規化 Lomb-Scargle ピリオドグラムのスペクトルを示す。スペクトルの横軸は周期の逆数 [cycle/day]、縦軸は正規化 Lomb-Scargle ピリオドグラムの値を示している。図 3 に示される正規化 Lomb-Scargle ピリオドグラムの最大値は $\omega = 0.039927[\text{cycle/day}]$ に出現しており、見積もりの正確さがわかる。

2.4 Lomb-Scargle ピリオドグラムによって与えられる独立周波数の数

等間隔の観測データのピリオドグラムによって求まる独立周波数の数 N は、式 (5) で与えられる通り、 $N = N_0/2$ である。しかし、不等間隔の観測データの Lomb-Scargle ピリオドグラムによって求まる独立周波数の数 N は、必ずしも式 (5) のようにはならない。そこで、 N を求めるためにいくつかの議論がある。例えば、はじめに Horne & Baliunas(1986) は、正規分布する雑音から成るデータを用いた Monte Carlo 法により、 N をデータ数 N_0 に対する二次関数で表した。しかし、Cumming, Marcy & Butler(1990) や Koen(1990), Schwarzenberg-Czerny(1996) などは、この結果は間違いであると指摘し、それぞれ異なる N と N_0 の関係を示した。いずれにせよ、独立周波

数の数 N を決定する正確な方法はないが、彼らは、データ数 N_0 が非常に小さい場合 ($N_0 \leq \sim 20$) を除いて $N \geq N_0/2$ であると結論付けている。また、データがほぼ等間隔の場合や、オーバーサンプリングが十分になされる場合、 N は $N_0/2$ に非常に近づくことが言える。したがって、本研究では、Lomb-Scargle ピリオドグラムによって求まる独立周波数の数 N は、式 (5) と同じ、 $N_0/2$ であると仮定する。

2.5 Lomb-Scargle ピリオドグラムの統計分布

Lomb-Scargle 法により、信号に含まれる周期を数値として求めることができる。しかし、Lomb-Scargle ピリオドグラムを計算するだけでは、その最大値がどれだけ有意なものなのか分からない。そこで、Lomb-Scargle ピリオドグラムの統計分布を調べることで、Lomb-Scargle ピリオドグラムの最大値の有意性を評価する。この節では、Lomb-Scargle ピリオドグラムの統計分布を仮説検定を用いて定量化する。

2.5.1 統計学における仮説検定

統計学の仮説検定では、帰無仮説 (記号は H_0) と、その逆の対立仮説 (記号は H_1) を立てる。帰無仮説には、検定で棄却したいものを設定する。帰無仮説を棄却することで、その内容が矛盾していると証明する。一方、対立仮説とは、帰無仮説が棄却されたときに採択される仮説である。注意したいのは、対立仮説の採択とは、帰無仮説が棄却できるという判断に過ぎないことである。統計学の仮説検定では、帰無仮説の下での統計量が、どれくらい珍しいかを測る基準の確率として有意水準 (記号は α) を用いる。

帰無仮説が正しいときに誤って棄却することを第 1 種の過誤といい、対立仮説が正しいときに誤って対立仮説を棄却 (帰無仮説を採択) することを第 2 種の過誤という。第 1 種の過誤の確率は、有意水準 α に等しい。一方、第 2 種の過誤の確率は β で表される。検定では、対立仮説が正しいときに対立仮説を採択する確率の大きさ $1 - \beta$ を測る。この $1 - \beta$ を検出力という。表 4 に帰無仮説 H_0 と対立仮説 H_1 の場合の各判断をまとめた。なお、() 内に書かれた言葉は信号検出理論の観点での表記である。第 1 種の過誤、有意水準、誤警報確率 (false alarm probability) は、すべて

表 4 各仮説の場合と判断

場合/判断	H_0 を採択	H_0 を棄却
	H_1 を棄却	H_1 を採択
H_0	正しい判断 (correct rejection)	第 1 種の過誤 α (false alarm)
H_1	第 2 種の過誤 β (miss)	正しい判断 [検出力] $1 - \beta$ (hit)

同じ意味である。

帰無仮説を棄却と判断する統計量の集合を棄却域といい、帰無仮説が棄却できない領域を採択域という。 α は帰無仮説が正しい状況で統計量が棄却域に入る確率、 β は対立仮説が正しい状況で統計量が採択域に入る確率とも表現できる。棄却域を広く取れば α は大きくなるが、同時に採択域が狭くなり β は小さくなる。 α を大きくして β を小さくすると、検出力 $1 - \beta$ が大きくなるため対立仮説は採択されやすくなるが、その信頼度 (有意性) は低くなってしまう。したがって、 α と β はトレードオフの関係にある。

本研究では、帰無仮説には「観測値は正規分布する雑音のみである ($X(t_i) = R(t_i)$)」を設定し、対立仮説には「観測値は雑音以外である ($X(t_i) \neq R(t_i)$)」を設定する。上述の通り、帰無仮説を棄却して対立仮説の採択することは、観測値が「正規分布する雑音以外」と判断することであり、必ずしも「信号 ($X(t_i) = X_S(t_i)$)」であるとは言い切れない。繰り返しになるが、§2.1 節で定義する雑音は、非周期的な強度変動や変動を示さない強度を含む恐れがあるため、雑音が正規分布するとは言い切れないことは念頭に置いておく必要がある。したがって、対立仮説が採択されても「正規分布する雑音以外」とであると判断しただけであって、正規分布に従わない非周期的な強度変動や変動を示さない強度は考慮されていないことに注意すべきである。

2.5.2 Lomb-Scargle ピリオドグラム の最大値の有意性と検出力

この節は、Scargle(1982) の §3 を参考に説明していく。

まず、帰無仮説 H_0 「観測値は正規分布する雑音のみである」の場合を考える。このとき、Lomb-Scargle ピリオドグラムは指数分布するので、確率変数を $Z = P_X(\omega)$ とすると、ある値 z に対する確率密度関数 $f_z(z)$ は、

$$f_z(z) = \Pr\{z < Z < z + dz\} = \exp(-z)dz$$

で与えられる。ただし、雑音の分散 σ_0^2 が 1 の場合である。ここで、 $\Pr\{\cdot\}$ は、 $\{\cdot\}$ が起こる確率を表す。確率分布関数 $F_z(z)$ は確率密度関数 $f_z(z)$ の積分で与えられるため、帰無仮説下での Lomb-Scargle ピリオドグラムの確率分布関数は、

$$F_z(z) = \Pr\{Z < z\} = \int_0^z f_z(z')dz' = 1 - \exp(-z)$$

となる。しかし、興味があるのは Lomb-Scargle ピリオドグラムの最大値である。そのため、 $P_X(\omega)$ が $\omega = \omega_{n_0}$ で最大値をとる場合を考え、確率変数を新たに $Z = P_X(\omega_{n_0})$ とおいて、独立周波数の数 N にわたって評価する。このとき、帰無仮説下での Lomb-Scargle ピリオドグラムの最大値の確率分布関数 $F_N(z)$ は、

$$F_N(z) = \Pr\{Z < z\} = [1 - \exp(-z)]^N$$

で与えられ、第 1 種の過誤 α は、

$$\alpha = 1 - F_N(z) = \Pr\{z < Z\} = 1 - [1 - \exp(-z)]^N \quad (11)$$

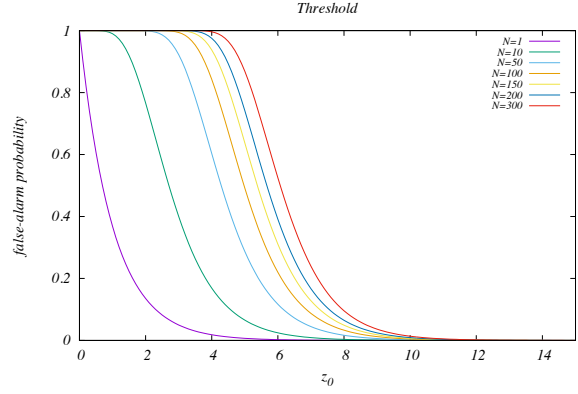
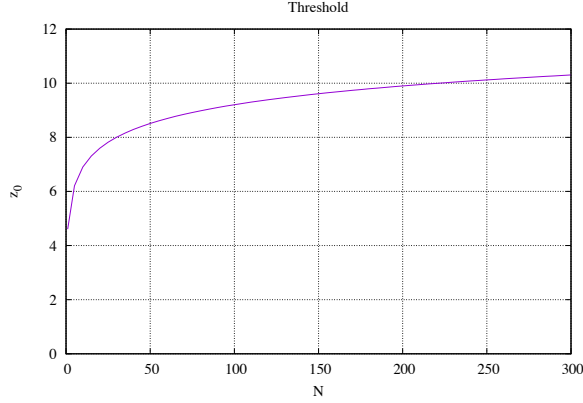


図4 誤警報確率 $\alpha = 0.01$ のときの式 (12) の様子 図5 様々な独立周波数 N と誤警報確率 α の関係

で与えられる。表4より、第1種の過誤 α は、信号検出理論の観点では「雑音を信号である」と判断してしまう誤警報確率 (false alarm probability) である。 α の値が小さいほど、Lomb-Scargle ピリオドグラム の最大値の信頼性は高いと言える。ここでの議論は、信号検出理論の観点を持つので、以下では、第1種の過誤を誤警報確率 (いずれも記号は α) と呼ぶことにする。

信号検出理論では、誤警報確率 α を例えば 0.01 や 0.05 などの小さな値で固定し、そのときの z の値をスペクトル図上に表した方が、Lomb-Scargle ピリオドグラム の最大値の信頼性の評価がしやすくなる。そのため、式 (11) を $z = z_0$ について解き、

$$z_0 = -\ln \left[1 - (1 - \alpha)^{1/N} \right] \quad (12)$$

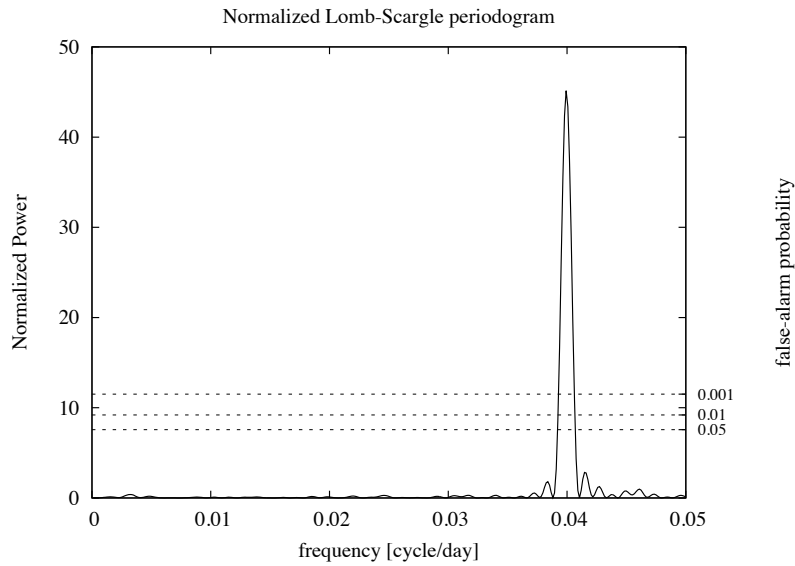


図6 図3に、誤警報確率 $\alpha = 0.001, 0.05, 0.01$ としたときの式 (12) で計算される閾値を加えたもの。

となる z_0 を、誤警報確率 α のときの閾値 (threshold) として扱うことにする。すなわち、Lomb-Scargle ピリオドグラム of 最大値が z_0 を下回った場合、信号は検出されなかったと明確に判断する基準を設けることに相当する。

図 4 は、誤警報確率 $\alpha = 0.01$ のときの式 (12) をプロットしたものである。 $N(= N_0/2)$ が大きくなるにつれて z_0 は大きくなるが、 N が大きすぎると z_0 は変化しづらくなることが分かる。一方、図 5 は、いくつかの場合で N を固定したときの閾値 z_0 に対する誤警報確率の変化の様子である。また、図 6 は、図 3 のスペクトル図に、いくつかの誤警報確率 α を閾値として設けたものである。参考例として示した正規化 Lomb-Scargle ピリオドグラムの最大値は $\alpha \ll 0.001$ となっており、非常に高い信頼度を持っていることがわかる。

次に、対立仮説 H_1 での Lomb-Scargle ピリオドグラムの統計分布を考える。ここでは、確率変数を、信号と雑音それぞれの Lomb-Scargle ピリオドグラムの比 (SN 比) $P_{\text{SN}}(\omega) = P_{\text{signal}}(\omega)/P_{\text{noise}}(\omega)$ とする。ある SN 比 p に対する確率分布関数 $F_{\text{SN}}(p)$ は、Groth(1975) で与えられる式を利用すると、

$$F_{\text{SN}}(p) = \Pr\{P_{\text{SN}} < p\} = 1 - \exp[-(p + P_{\text{SN}})]\phi(p, P_{\text{SN}})$$

で与えられる。ここで ϕ は第 1 種変形 Bessel 関数の積分であり、

$$\phi(p, P_{\text{SN}}) = \sum_{m=0}^{\infty} \sum_{k=0}^m \frac{p^k P_{\text{SN}}^m}{k!m!}$$

である。

それでは、対立仮説が正しいのに棄却をしてしまう第 2 種の過誤 β について考える。独立周波数の数 N にわたって、信号が存在する場合でも Lomb-Scargle ピリオドグラムの最大値が、式 (12) によって求まる閾値 z_0 を下回った場合、信号は検出されなかったと判断するので β は、

$$\beta = \Pr\{\text{miss}\} = 1 - \exp[-(z_{\text{SN}} + P_{\text{SN}})]\phi(z_0, P_{\text{SN}}) \quad (13)$$

で与えられる。したがって、検出力 $1 - \beta$ は、

$$1 - \beta = \Pr\{\text{hit}\} = \exp[-(z_{\text{SN}} + P_{\text{SN}})]\phi(z_0, P_{\text{SN}})$$

で与えられる。ここで、 z_{SN} は、閾値 z_0 での SN 比である。

図 7 に、誤警報確率 $\alpha = 0.01$ のときの、各 N についての検出力 $1 - \beta$ の変化を示した。図 7 より、 P_{SN} が大きくなるにつれて検出力が大きくなることがわかる。また、 N が小さいほど検出力は大きくなることがわかる。

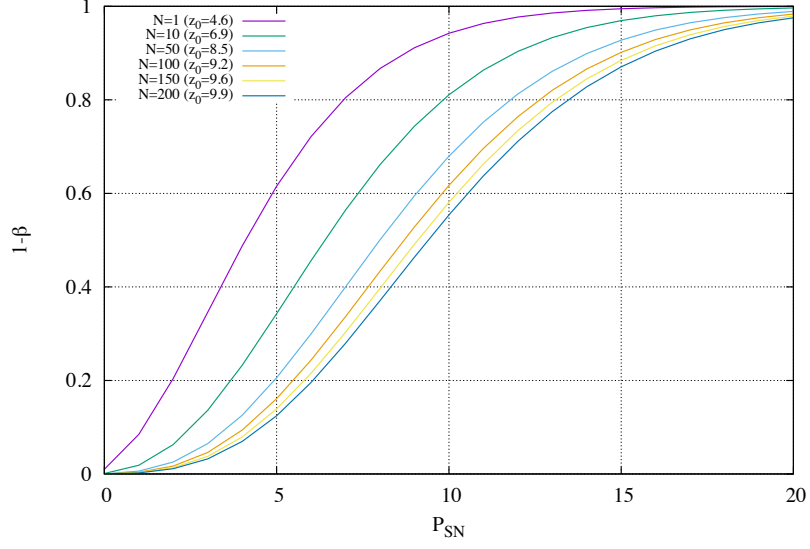


図 7 誤警報確率 $\alpha = 0.01$ のときの、各 N についての検出力 $1 - \beta$ の変化

正規化 Lomb-Scargle ピリオドグラムの SN 比は、

$$P_{N,\text{SN}}(\omega) = \frac{P_{N,\text{signal}}(\omega)}{P_{N,\text{noise}}(\omega)} = 1$$

となることが期待される。なぜなら、正規化とは、データに信号が含まれる場合であっても、データが標準正規分布 $N(0, 1)$ に従う雑音だけの場合に求まる Lomb-Scargle ピリオドグラムと同等に扱えるようにするための行為だからである。正規化することによって、上述の帰無仮説を用いた誤警報確率を信頼度の指標として扱えることになる。一方、正規化を行わない通常の Lomb-Scargle ピリオドグラムは、検出力の導出に用いることができる。しかし、実際の観測データから信号と雑音を切り離すことは難しいため、本研究では検出力を求めることはしない。

3 解析プログラム

この章では、正規化 Lomb-Scargle ピリオドグラムをコンピュータを用いて計算する手法について説明する。本研究では、正規化 Lomb-Scargle ピリオドグラムを計算するために、C 言語プログラム `lsperiodogram` を作成した (付録 D)。正規化 Lomb-Scargle ピリオドグラムを計算するためのアルゴリズムについて参考にしたのは、Numerical Recipes in C (Press et al. 1988) の §13.8 Spectral Analysis of Unevenly Sampled Data である。正規化 Lomb-Scargle ピリオドグラムの計算に当たる C 言語プログラムの関数^{*5} は、`lsperiodogram` 中の関数 `period` である。

3.1 日立アンテナ・モニター観測

計算対象のデータは、茨城県日立市にある日立アンテナを用いた、6.7GHz メタノールメーザーのモニター観測によって得られるものである。モニター観測とは、ある期間をかけて天体を観測し、データを得る観測である。日立アンテナでのモニター観測は、3つの期間に分かれており、第1期は2012年12月30日-2014年1月10日、第2期は2014年5月7日-2015年8月24日、第3期は2015年9月18日-である。観測対象の6.7GHz メタノールメーザーは、赤緯-30度以北の461天体、観測日間隔は約9-10日である。ただし第3期では、より短周期な変動を観測するため、強度変動が顕著な143天体を選出し、約4-5日間隔で観測している。また、一部の天体は較正目的などで毎日観測を行っているものもある。日立アンテナでのモニター観測のパラメータを表5にまとめた。

表5 モニター観測の概要

項目	値
アンテナ	日立 32m 電波望遠鏡
期間	第1期：2012/12/30 - 2014/1/10 第2期：2014/5/7 - 2015/8/24 第3期：2015/9/18 -
観測周波数帯	6664 – 6672 [MHz]
空間分解能	~ 4.6 [arcmin]
速度分解能	0.044 [km/s]
偏波	LHCP
積分時間	300 [s]
最小検出感度	~ 0.9 [Jy]

^{*5} C 言語では、あるまとまりの計算や処理のみを行う機能を関数と呼ぶ。

3.2 データの前処理

3.2.1 トレンド成分の除去

データにトレンド成分が含まれる場合、統計的性質は時不変でなくなるため、定常性がなくなる。したがって、トレンド成分が含まれるデータの正規化 Lomb-Scargle ペリオドグラムは正しく計算されない。そのため、データからトレンド成分を取り除く前処理が必要がある。本研究で取り扱う観測データはトレンド成分が含まれていると仮定しているため、プログラム `lsperiodogram` にはトレンド除去の処理を入れている。また、含まれるトレンド成分は、1 次関数であると仮定している。プログラムでは、トレンド成分の 1 次関数を求めるために、観測データ $X(t_i)$ を最小二乗法を用いて

$$x(t_i) = mt_i + l \quad (14)$$

にフィットさせる。ここで、 m , l は定数である。その後、求めた 1 次関数の傾き成分 mt_i を観測データから引く ($X(t_i) - mt_i$) ことによりトレンド除去を行う。

具体的には、最小二乗法を用いて N_0 個の観測データ $(t_1, X(t_1)), (t_2, X(t_2)), \dots, (t_{N_0}, X(t_{N_0}))$ を式 (14) にフィットさせるためには、

$$J = \frac{1}{2} \sum_{i=1}^{N_0} (x(t_i) - (mt_i + l))^2$$

で与えられる J の最小値を求める。したがって、 $\partial J / \partial m = 0$, $\partial J / \partial l = 0$ となる、 m , l を求めればよい。そのため、プログラムでは、

$$\begin{aligned} \frac{\partial J}{\partial m} &= \sum_{i=1}^{N_0} (x(t_i) - mt_i - l)(-t_i) = m \sum_{i=1}^{N_0} t_i^2 + l \sum_{i=1}^{N_0} t_i - \sum_{i=1}^{N_0} t_i x(t_i), \\ \frac{\partial J}{\partial l} &= \sum_{i=1}^{N_0} (x(t_i) - mt_i - l)(-1) = m \sum_{i=1}^{N_0} t_i + l \sum_{i=1}^{N_0} 1 - \sum_{i=1}^{N_0} x(t_i) \end{aligned}$$

より、

$$\begin{pmatrix} \sum_{i=1}^{N_0} t_i^2 & \sum_{i=1}^{N_0} t_i \\ \sum_{i=1}^{N_0} t_i & \sum_{i=1}^{N_0} 1 \end{pmatrix} \begin{pmatrix} m \\ l \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{N_0} t_i x(t_i) \\ \sum_{i=1}^{N_0} x(t_i) \end{pmatrix}$$

を解く `fit` 関数を与える。求まった m と l を式 (14) に代入し、`main` 関数内でトレンド除去 $X(t_i) - mt_i$ を行う。今後の説明のため、トレンド除去したデータを改めて、

$$h(t_i) = X(t_i) - mt_i \quad (15)$$

で表す。

3.2.2 データの平均を 0 にする

§2.1 節で定義する雑音は、平均が 0 である正規分布 $N(0, \sigma_0^2)$ に従うと仮定している。そのため、2 つ目の前処理として、トレンド除去した時系列の平均を 0 にする。具体的には、トレンド除去を行ったデータ $h(t_i)$ から、その平均

$$\bar{h} = \frac{1}{N_0} \sum_{i=1}^{N_0} h(t_i)$$

を引く。すなわち、プログラムで計算する正規化ピリオドグラムは、

$$P'_N(\omega) = \frac{1}{2\sigma_0^2} \left\{ \frac{\left[\sum_{j=1}^{N_0} (h(t_j) - \bar{h}) \cos \omega(t_j - \tau) \right]^2}{\sum_{j=1}^{N_0} \cos^2 \omega(t_j - \tau)} + \frac{\left[\sum_{j=1}^{N_0} (h(t_j) - \bar{h}) \sin \omega(t_j - \tau) \right]^2}{\sum_{j=1}^{N_0} \sin^2 \omega(t_j - \tau)} \right\} \quad (16)$$

となる。本研究では、観測データに対して式 (16) で表される正規化 Lomb-Scargle ピリオドグラムを計算することを解析と呼ぶことにする。

3.3 オーバーサンプリング

§2.2 節で説明したオーバーサンプリング (補間) は、Lomb-Scargle ピリオドグラムにも適用可能である。§2.2 節では、周波数分解能を細かくするために、 N_0 個のデータを m 倍にオーバーサンプリングすると、独立周波数の数は $mN_0/2$ 個になると説明した。また、等間隔に並べられたデータに含まれる周期関数 $X_S(t_i)$ の周波数 ω_{n_0} が、ナイキスト周波数を越える場合、エイリアシングと呼ばれる現象により、一般には正しいピリオドグラムが導出できなくなる。この現象を考慮することやナイキスト周波数以上を調べたい場合のため、プログラム `lsperiodogram` では、導出周波数をナイキスト周波数の何倍まで求めたいかを指定できるようにしている。そのため、プログラム `lsperiodogram` では、独立周波数の数を次のように制御する。

まず、プログラム `lsperiodogram` では、導出される独立周波数の分解能を上げたい場合、引数 `ofac` を用いることにする。引数 `ofac` は、§2.2 節で与えた m に相当する。このとき導出される独立周波数の数 N_{out} は、

$$N_{\text{out}} = \text{ofac} \times \frac{N_0}{2}$$

である。一方、何倍のナイキスト周波数まで独立周波数を求めたいかを制御するために引数 `hifac` を与える。このとき、ナイキスト周波数 ω_c に対し、周波数 ω_{hi} まで求めたい場合、引数 `hifac` は、

$$\text{hifac} = \frac{\omega_{hi}}{\omega_c}$$

で与えられる。したがって、プログラム `lsperiodogram` で導出する独立周波数の数 `nout` は、

$$\text{nout} = \frac{\text{ofac} \times \text{hifac}}{2} N_0 \quad (17)$$

となる。例えば、4 倍にオーバーサンプリング (周波数分解能を $1/4$ にしたい), かつ 2 倍のナイキスト周波数まで調べたい場合, $\text{ofac} = 4$, $\text{hifac} = 2$ と設定すればよい。

4 結果・考察

日立アンテナ・モニター観測対象の 461 天体の内、解析可能なデータ^{*6}がある 443 天体 (付録 E) をプログラム `lsperiodogram` で解析した結果を示す。解析対象のデータは、3 つの期間を合わせたものであり、日立アンテナでモニター観測が開始された 2012 年 12 月 30 日から 2016 年 1 月 3 日現在までで区切っている。2016 年 1 月現在、モニター観測は続いているので、暫定的な結果を示すことになる。ここでは、条件を課すことによって選ばれた、新検出の周期変動天体 11 天体と、既知の周期変動天体の内、新たに異なる周期が導出された 1 天体を紹介する。

4.1 選出条件

一つの天体につき、いずれかの速度成分が、

1. 正規化 Lomb-Scargle ピリオドグラム (式 (16)) によって導出された最大値 $P'_N(\omega_p)$ が、誤警報確率 $\alpha \leq 0.001$ であること。
2. 条件 1. で選出された周期 $1/\omega_p$ が、天体の全観測期間に 3 周期含まれるもの。
3. オーバーサンプリング (補間) を行わない場合 ($\text{ofac} = 1$) と、オーバーサンプリングを行う場合 ($\text{ofac} = 8$) の両方で条件 1., 2. を満たすこと。
4. 天体自体が明るく、かつ強度変動していると思われること。

の全てを満たす場合、周期的な強度変動天体であると判断する。

条件 2., 3. 及び 4. を課すのは、§2.1 節で定義する信号と雑音の定義が大雑把であることによる弊害を考慮する必要があるためである。§2.1 節で定義する雑音は、正規分布 $N(0, \sigma_0^2)$ に従うと仮定しているが、必ずしもそう言い切れる保証はない。サンプリング定理から、観測日間隔 Δt が等間隔である場合、導出可能な最小周期は $2\Delta t$ であり、最大周期は観測期間に等しい。しかし、非周期的な強度変動や変動を示さない天体の強度は連続スペクトルとして $\omega = 0$ 以外でも大きな値をとる恐れがある。したがって、導出された周期が観測期間に複数回 (ここでは 3 回) 確認できること (条件 2.), 観測期間中に十分に強度変動していると判断できること (条件 4.) を確認する。強度変動の有無を判断する指標として、各速度成分に対し、全観測値の標準偏差 σ を平均 μ で割った、

$$I_{\text{mod}} = \frac{\sigma}{\mu},$$

を変動係数 I_{mod} と定義する。選出条件では、 $I_{\text{mod}} \geq 0.3$ を満たす場合、天体の速度成分は強度変動していると判断する。また、電波天文観測で与えられる天体からの強度は、非常に微弱なため、必ずしも十分な SN 比^{*7} が保たれるわけではない。そのため、ある程度、強度を持つ天体であるかを考慮する必要もある。選出条件では、観測期間中の強度の最大値 S_{max} と観測装置由来の雑音の

^{*6} データが付録 D に記した書式で記録されたもの。

^{*7} ここで言う SN 比は、天体からの電波強度と観測装置に由来する雑音との比である。

rms 値 σ の比, 最大強度対装置雑音比 $S_{\max}/1\sigma \geq 7$ (すなわち, 7σ 以上) であるものを十分な強度で観測された天体であると判断する。

条件 3. を課すのは, §2.5.2 で与えられる議論に基づいている。本研究では, 検出力は, 対立仮説「観測値は雑音以外である」を採択する確率であると定義している。また, 独立周波数の数 N が大きくなるとともに, 検出力は小さくなることに注意する。つまり, オーバーサンプリングをすることは, 独立周波数の数 N を大きくすることになるため, 検出力が小さくなることに注意する必要がある。もちろん, 独立周波数の数が増えたことで, オーバーサンプリングを行わない場合では見つからなかった周波数に最大値が現れることが考えられる。実際, 観測データの中には, オーバーサンプリングを行った場合のみ, 他の条件 (1., 2., 4.) を満たす天体が見つまっている。しかし, 本研究では雑音の定義が大雑把なため検出力が十分あるかどうかを考慮したい。したがって, オーバーサンプリングを行う場合と行わない場合の両方で, 正規化 Lomb-Scargle ピリオドグラムが最大値を持つのであれば, より細かい独立周波数でありながらも検出力が十分にある結果が得られたと判断することができる。なお, 条件を満たした場合, 採択する周期は $\text{ofac} = 8$ のときに導出されたものである。

4.2 新検出の周期強度変動天体

表 6 に, 既知の周期強度変動天体以外で全ての選出条件を満たす天体とその速度成分を示す。また, 表 7 に既知の周期強度変動天体の内, 全ての選出条件を満たすものであり, 大きく異なる周期が導出された天体とその速度成分を示す。表 6, 表 7 には, 各天体の詳細な観測期間とその通算日数を示してある。通算日数 (観測期間) T は, 導出可能な最大周期や独立周波数を計算するために示してある。また, ここで紹介する天体は, $\text{ofac} = 8$, $\text{hifac} = 1$ で統一しているため, 式 (17) で求まる独立周波数の数 nout は, $4N_0$ である。

4.2.1 G 004.43+00.13

表 8 に, G 004.43+00.13 の各視線速度成分の変動係数 I_{mod} , $S_{\max}/1\sigma$, 正規化ピリオドグラムの最大値, そのときの誤警報確率を示す。また, 選出条件の内, 「1. 誤警報確率 $\alpha \leq 0.001$ であること」と「2. 周期 $1/\omega_p$ が, 天体の全観測期間に 3 周期含まれるもの」を満たす速度成分のみ, 導出周期を示す。混乱を避けるため, 全ての選出条件を満たす速度成分には, 周期の横に (*) を付けている。図 8*⁸ には, G 004.43+00.13 の正規化ピリオドグラムのスペクトルを示す。図 8 に示されるスペクトルは, 式 (16) によって求まる値を線で結んだものであり, 実際は離散スペクトルである。また, 選出条件をすべて満たす速度成分が 1 つだけの場合, その成分は赤線で示している。この正規化 Lomb-Scargle ピリオドグラムのスペクトルは, $\text{ofac} = 8$ のときに導出されたものである。図 9 には, 全速度成分のフラックス密度を観測期間にわたって積分したラインプロファイルを示す。さらに, 図 9 には, 周期強度変動している速度成分上に縦線を挿入している。図 10 に

*⁸ 図のレイアウトは, 見やすくするためにプログラム `lsperiodogram` で出力される作図用ファイルの一部を修正している。特に, 周波数軸 (横軸) は, 求まる周波数のすべてを図示しているわけではない。

表 6 新検出の周期強度変動天体

天体名	観測期間	通算日数 [day]	データ数 N_0	視線速度 V_{lsr} [km/s]	周期 T [day]
G 004.43+00.13	2013/01/02 - 2016/01/02	1096	76	0.91	365.05
G 006.18-00.35	2013/01/02 - 2015/12/28	1091	90	-35.69	181.67
G 014.23-00.50	2013/01/04 - 2016/01/02	1094	470	25.30	23.96
G 024.14+00.00	2012/12/30 - 2015/12/29	1095	85	19.28	186.21
G 026.52-00.26	2013/01/01 - 2016/01/03	1095	91	109.58	178.94
G 030.40-00.29	2013/01/04 - 2015/12/29	1090	87	105.03	223.39
G 033.64-00.21	2013/01/06 - 2016/01/03	1093	468	58.94	189.94
G 034.39+00.24	2013/01/06 - 2015/12/31	1090	105	55.67	29.33
				60.46	29.33
G 036.70+00.09	2013/01/02 - 2016/01/03	1097	92	53.11	53.46
				53.65	53.14
				55.09	53.46
				62.10	53.46
G 037.47-00.11	2013/01/07 - 2015/12/31	1089	85	54.51	68.54
G 174.19-00.09	2012/12/30 - 2016/01/03	1100	510	2.04	46.81

表 7 既知の周期変動天体の内、新たに異なる周期が導出された天体

天体名	観測期間	通算日数 [day]	データ数 N_0	視線速度 V_{lsr} [km/s]	周期 T [day]
G 196.45-01.67	2012/12/30 - 2016/01/03	1100	367	14.70	146.53

は、選出条件をすべて満たす速度成分の各観測日のフラックス密度の変化の様子を示す。さらに、図 10 には、観測データ第 1 点目から導出周期毎に区切った格子線を挿入している。以下で紹介する 10 天体についても表と図の内容は同じであるため説明は省略する。

G 004.43+00.13 は 16 個の速度成分が同定されている。この内、5 つの速度成分が、誤警報確率 $\alpha \leq 0.001$ である。視線速度 $-0.95[\text{km/s}]$ の成分は導出周期が 876.12 日、視線速度 $0.53[\text{km/s}]$, $1.29[\text{km/s}]$, $1.92[\text{km/s}]$ の成分は 380.92 日である。この 4 つの速度成分は変動係数 I_{mod} が選出条件の 0.3 以上であるが、視線速度 $0.53[\text{km/s}]$, $1.29[\text{km/s}]$ の成分は、最大強度対装置雑音比

表 8 G 004.43+00.13 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
-5.56	0.525	2.2	4.16	0.991786	-
-4.53	0.438	4.9	5.06	0.856966	-
-1.7	0.715	2.8	4.6	0.953063	-
-0.95	0.831	24.8	18.39	0.000003	876.12
-0.23	0.566	3.4	4.62	0.950576	-
0.1	0.321	5.8	5.04	0.860956	-
0.53	0.338	6.9	16.65	0.000018	380.92
0.91	0.334	8.7	16	0.000034	365.05(*)
1.29	0.348	6.1	13.78	0.000315	380.92
1.92	0.328	8.3	19.68	0.000001	380.92
3.03	0.576	3.5	6	0.531206	-
3.58	0.421	3.2	3.37	0.999975	-
4.52	0.341	3.4	6.84	0.278059	-
5.71	0.408	6.4	5.02	0.865476	-
7.53	0.557	7.2	11.89	0.002092	-
8.77	0.448	3.9	6.21	0.455852	-

$S_{\text{max}}/1\sigma$ が選出条件の 7 以下である。また、観測期間が 1096 日のため、導出周期 365.333 日以上は 3 周期分の観測ができていない。したがって、新検出として選出していない。一方、選出条件のすべてを満たす視線速度 0.91[km/s] の成分は、導出周期が 365.05 日であり、隣合う独立周期との差は +15.87 日、−14.54 日である。この隣合う独立周期とは、8 倍にオーバーサンプリング (ofac = 8) した際に求まる独立周波数において、正規化 Lomb-Scargle ピリオドグラムが最大値をとる周波数の両隣の周波数の逆数のことである。すなわち、導出周期の誤差とも解釈できる。G 004.43+00.13 の導出周期 365.05 日の 3 周期分は、1095.15 日となるため観測期間 1096 日に僅差で収まっている。現時点では、誤差を考慮すると導出周期の 3 周期分が観測期間に収まらないことになる。ただし、今後もモニター観測は続いていくため、観測期間は増えてることになり導出周期の 3 周期分が観測期間に十分収まっていくことが期待される。

4.2.2 G 006.18-00.35

G 006.18-00.35 は、12 個の速度成分が同定されている。この内、3 つの速度成分が誤警報確率 $\alpha \leq 0.001$ かつ観測期間に 3 周期分を含んでいる。しかし、視線速度 −37.18[km/s] と −36.20[km/s] の成分は、変動係数が 0.3 以下であるため、新検出として選出していない。選出条件のすべてを満たす視線速度 −35.69[km/s] の成分は、導出周期が 181.67 日であり、隣合う独立周期との差は +7.90 日、−3.71 日である。

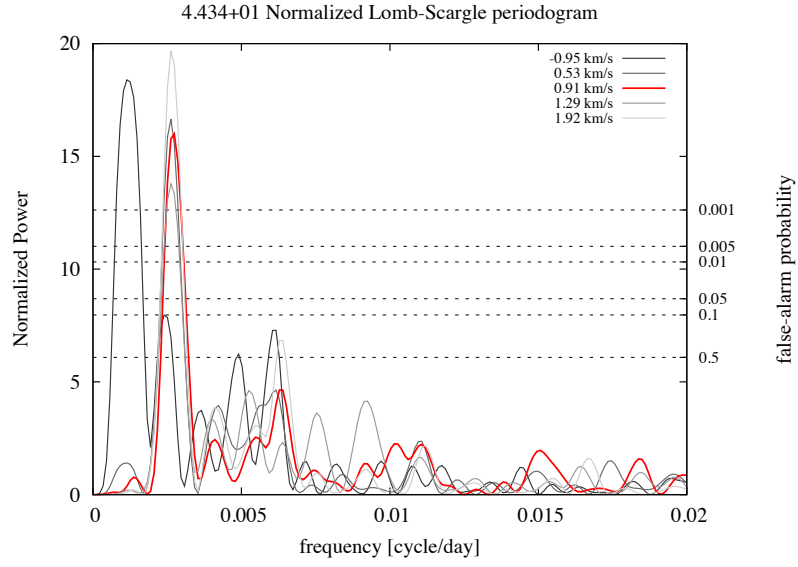


図 8 G 004.43+00.13 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

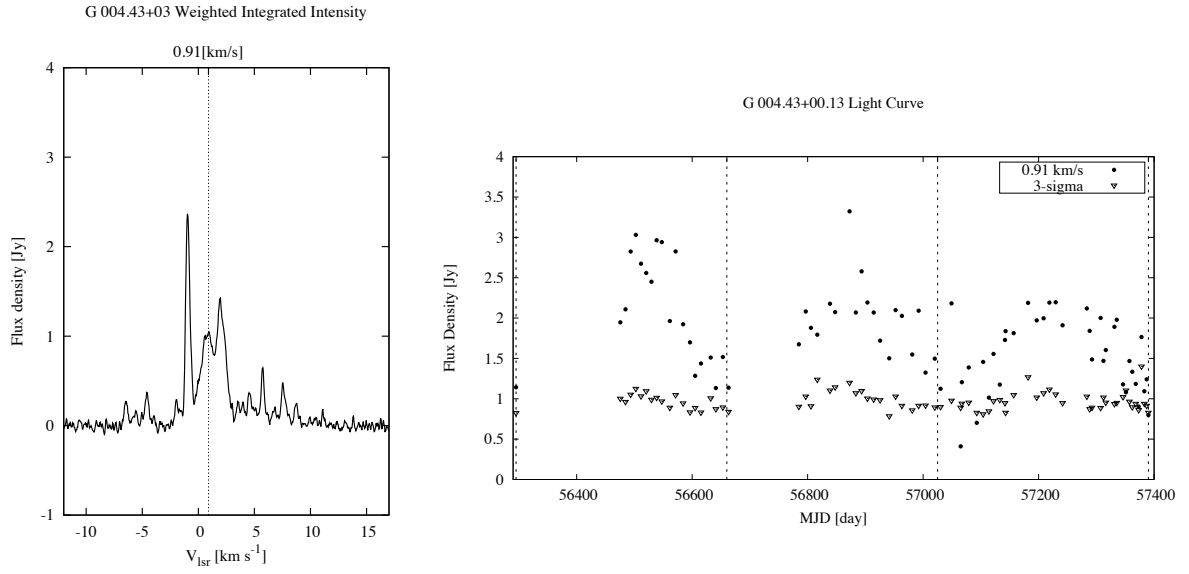


図 9 G 004.43+00.13 のライン
プロフィール

図 10 G 004.43+00.13 の強度変動時系列

表 9 G 006.18-00.35 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
-37.18	0.251	12.8	16.07	0.000017	181.67
-36.2	0.244	81.6	14.27	0.000168	185.54
-35.69	0.32	36.4	21.6	0.8×10^{-8}	181.67(*)
-34.48	0.19	31.7	12.35	0.000872	-
-34.12	0.145	21.2	6.32	0.491012	-
-32.68	0.242	28.8	7.96	0.056087	-
-32.12	0.194	14.9	7.08	0.303428	-
-31.57	0.18	40.2	7.25	0.165816	-
-31.32	0.157	28.6	6.14	0.593395	-
-30.17	0.147	650.8	5.16	0.755813	-
-28.58	0.183	52.1	11.68	0.005659	-
-26.49	0.443	3.2	4.55	0.975140	-

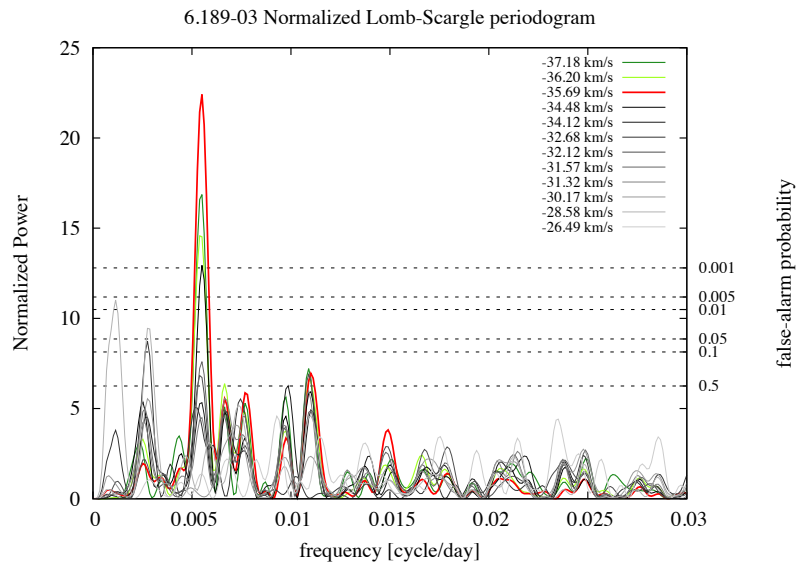


図 11 G 006.18-00.35 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

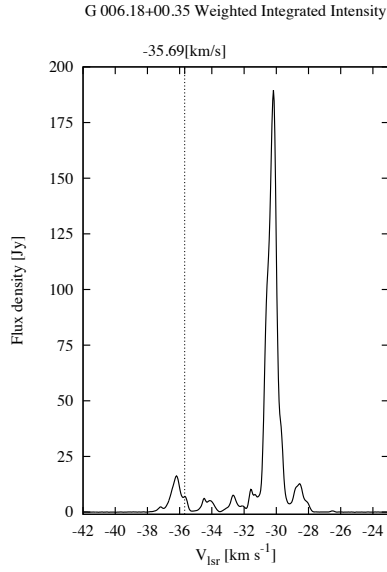


図 12 G 006.18-00.35 のライン
プロファイル

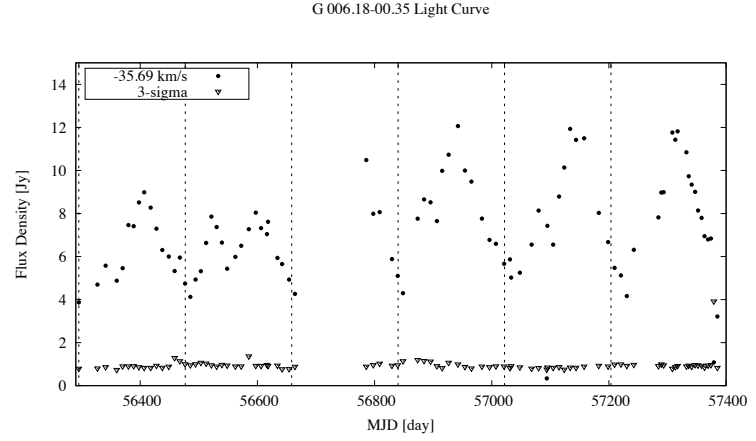


図 13 G 006.18-00.35 の強度変動時系列

4.2.3 G 014.23-00.50

G 014.23-00.50 は、9 個の速度成分が同定されている。この内、3 つの速度成分が誤警報確率 $\alpha \leq 0.001$ かつ観測期間に 3 周期分を含んでいる。しかし、視線速度 20.52[km/s] と 20.98[km/s] の成分は、最大強度対装置雑音比 $S_{\max}/1\sigma$ が選出条件の 7 以下であるため、新検出として選出していない。選出条件のすべてを満たす視線速度 25.30[km/s] の成分は、導出周期が 23.96 日であり、隣合う独立周期との差は +0.13 日、-0.07 日である。新検出の全 11 天体の内、最も短い周期が導出されている。正規化 Lomb-Scargle ピリオドグラムの独立周波数は等間隔に求まるため、周波数が大きいほど (短周期ほど)、隣合う周期の差が少なくなる。したがって、最も高精度に周期が導出されたことになる。さらに、G 014.23-00.50 のデータ数は 470 個と比較的多いことから、高精度な結果が得られたことが期待される。なお、この天体についての詳細な解析は、永瀬桂修士論文 (茨城大学; 2015 年度) で行なわれている。

表 10 G 014.23-00.50 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
20.52	0.673	4.4	21.21	0.000001	178.46
20.98	0.609	2.6	26.17	5.0×10^{-9}	323.86
21.32	0.822	2.6	8.12	0.404722	-
21.75	0.807	5.1	17.93	0.000120	-
22.3	1.679	24.1	45.11	0.0×10^{-19} 以下	-
23.46	1.539	15.9	39.76	0.0×10^{-19} 以下	-
23.91	0.958	13.3	58.65	0.0×10^{-19} 以下	-
24.69	0.737	7.4	56.87	0.0×10^{-19} 以下	-
25.3	1.364	23.8	40.63	0.0×10^{-19} 以下	23.96(*)

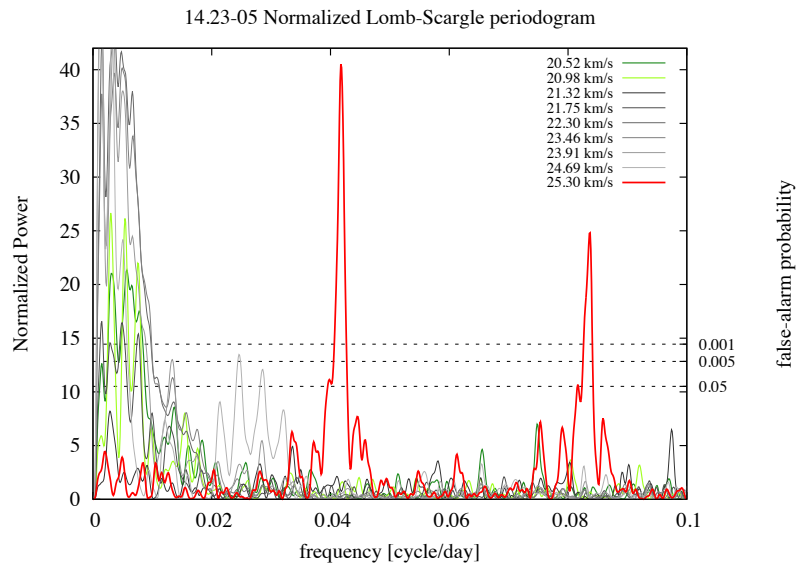


図 14 G 014.23-00.50 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

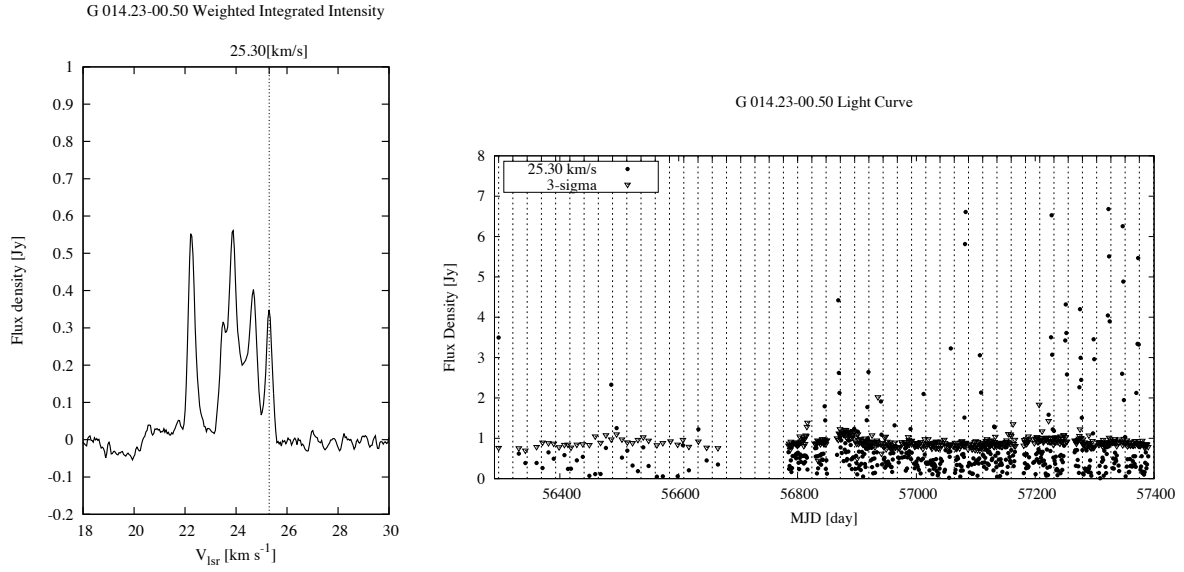


図 16 G 014.23-00.50 の強度変動時系列

図 15 G 014.23-00.50 のライン
プロフィール

4.2.4 G 024.14+00.00

G 024.14+00.00 は、3 個の速度成分が同定されている。この内、2 つの速度成分が誤警報確率 $\alpha \leq 0.001$ である。しかし、視線速度 17.52[km/s] の成分は、変動係数が 0.3 以下であるため、新検出として選出していない。選出条件のすべてを満たす視線速度 19.28[km/s] の成分は、導出周期が 186.21 日であり、隣合う独立周期との差は +4.04 日、-3.88 日である。

4.2.5 G 026.52-00.26 の解析結果

G 026.52-00.26 は、9 個の速度成分が同定されている。この内、選出条件のすべてを満たすのは、視線速度 109.58[km/s] の成分のみであり、導出周期は 178.94 日、隣合う独立周期との差は +3.73 日、-3.58 日である。誤警報確率 $\alpha \leq 0.001$ の速度成分が無いこと、変動係数 $I_{\text{mod}} \leq 0.3$ である速度成分が多いことから、視線速度 109.58[km/s] 以外の成分は、今後、観測データを増やしていても、周期性が確認される見込みは少ないと思われる。

表 11 G 024.14+00.00 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
17.52	0.124	87.3	13.44	0.000493	186.21
17.71	0.107	117.9	9.59	0.023053	-
19.28	0.332	12.4	27.09	5.8×10^{-10}	186.21(*)

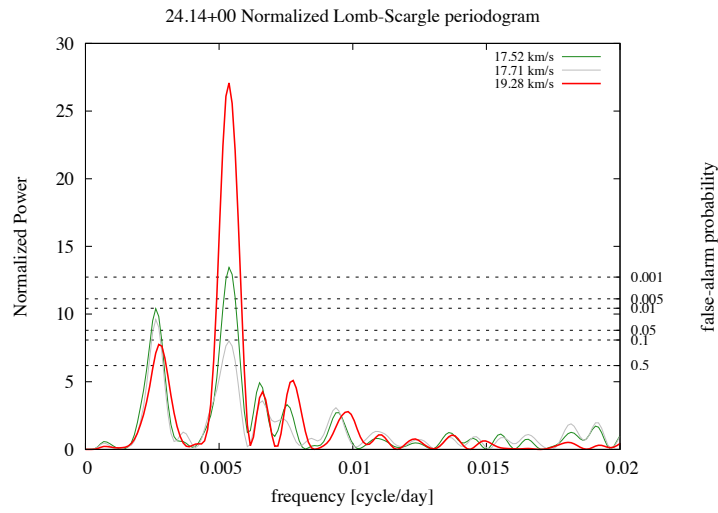


図 17 G 024.14+00.00 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

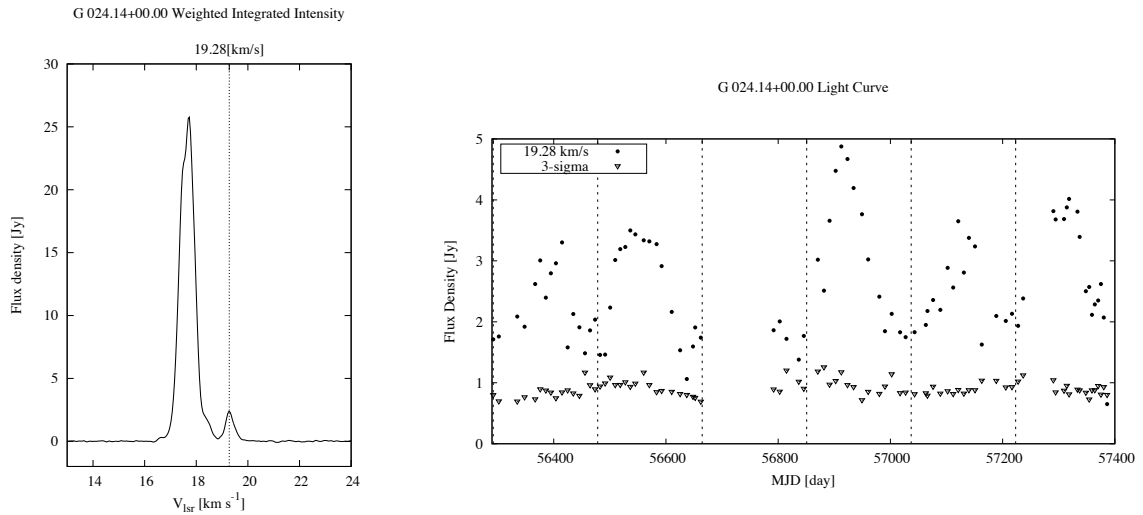


図 19 G 024.14+00.00 の強度変動時系列

図 18 G 024.14+00.00 のラインプロファイル

表 12 G 026.52-00.26 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
103.35	0.156	12.4	9.19	0.036343	-
103.8	0.141	15.9	6.66	0.371502	-
104.22	0.149	21.5	9.74	0.021216	-
105.38	0.27	4.6	5.54	0.760375	-
106.11	0.307	3.3	4.52	0.981585	-
107.1	0.295	5.7	3.79	0.999761	-
107.7	0.322	4.7	3.79	0.999763	-
108.27	0.257	5.5	4.66	0.968097	-
109.58	0.515	12.8	18.27	0.000004	178.94(*)

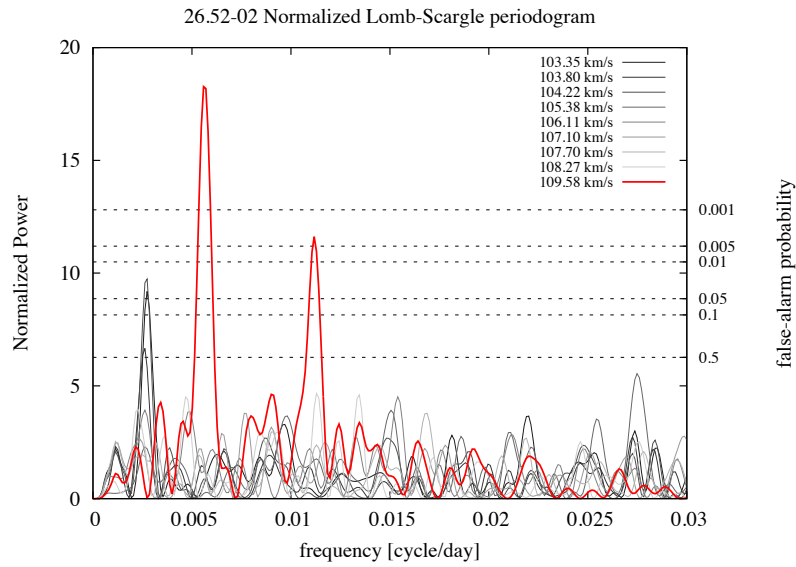


図 20 G 026.52-00.26 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

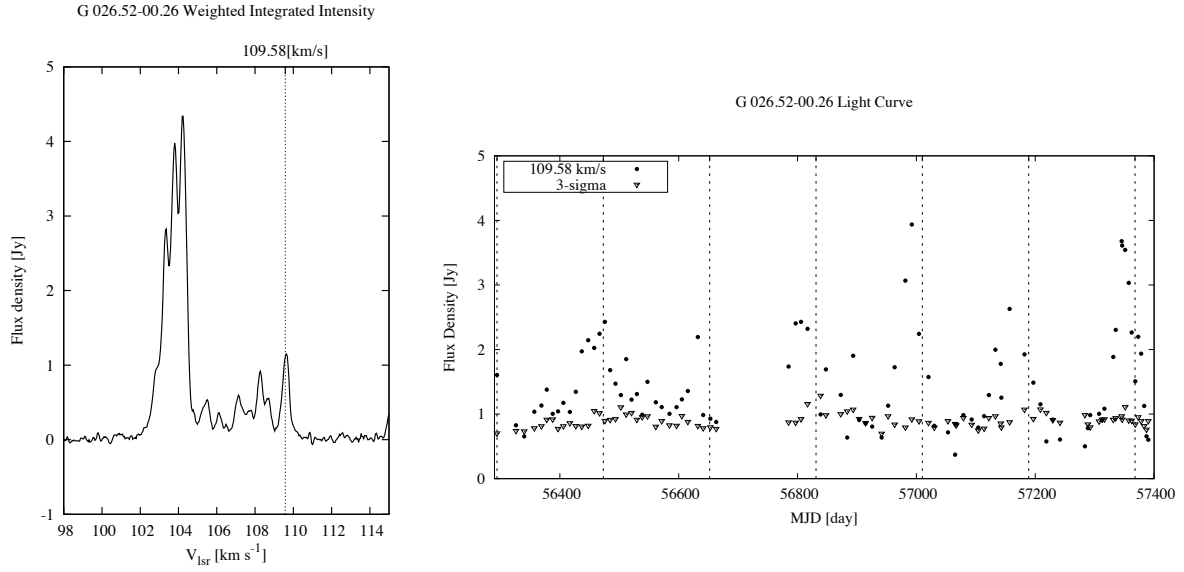


図 22 G 026.52-00.26 の強度変動時系列

図 21 G 026.52-00.26 のライン
プロファイル

4.2.6 G 030.40-00.29

G 030.40-00.29 は、9 個の速度成分が同定されている。この内、2 つの速度成分が誤警報確率 $\alpha \leq 0.001$ かつ観測期間に 3 周期分を含んでいる。この 2 つの内、視線速度 104.29[km/s] の成分は、最大強度対装置雑音比 $S_{\text{max}}/1\sigma$ が選出条件の 7 以下であるため、新検出として選出していない。しかし、最大強度対装置雑音比 $S_{\text{max}}/1\sigma = 6.6$ であるため、選出条件を少々変更すると、新検出と主張できなくはない。選出条件のすべてを満たす視線速度 105.03[km/s] の成分は、導出周期が 223.39 日であり、隣合う独立周期との差は +5.87 日、-5.59 日である。

4.2.7 G 033.64-00.21

G 033.64-00.21 は、8 個の速度成分が同定されている。この内、視線速度 57.04[km/s] を除く、7 つの速度成分が誤警報確率 $\alpha \leq 0.001$ である。この内、選出条件のすべてを満たす視線速度 58.94[km/s] の成分以外は、最大強度対装置雑音比が $S_{\text{max}}/1\sigma \gg 7$ であるものの、いずれかの条件を満たさない。視線速度 58.94[km/s] の成分の導出周期は 189.94 日であり、隣合う独立周期との差は +4.22 日、-4.04 日である。

表 13 G 030.40-00.29 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
98.16	0.246	18.2	10.76	0.007328	-
100.41	0.258	14.1	12.29	0.001602	-
101.31	0.557	3.3	3.28	0.999998	-
101.56	0.353	3.4	4.35	0.989035	-
101.91	0.279	4.1	5.26	0.837276	-
103.02	0.182	10.8	5.78	0.658301	-
103.67	0.207	9.2	7.68	0.149150	-
104.29	0.507	6.6	15.58	0.000060	217.8
105.03	0.647	29.1	29.6	4.9×10^{-11}	223.39(*)

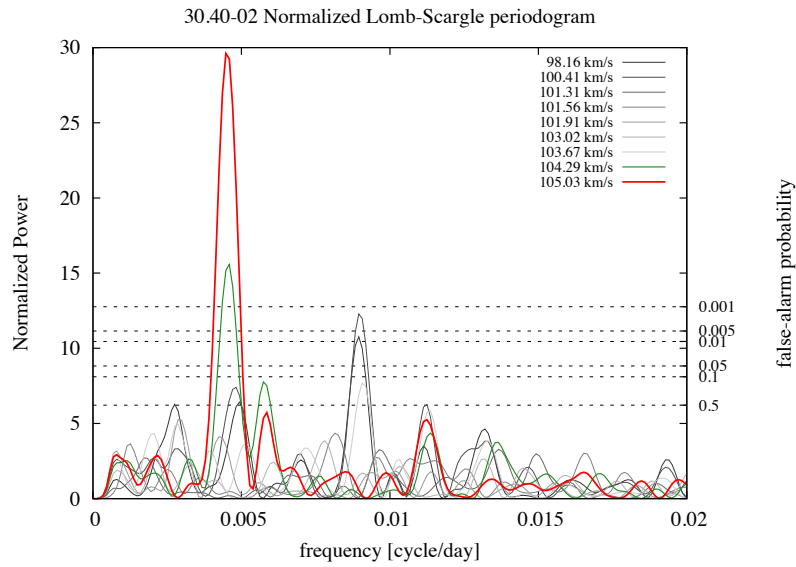


図 23 G 030.40-00.29 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

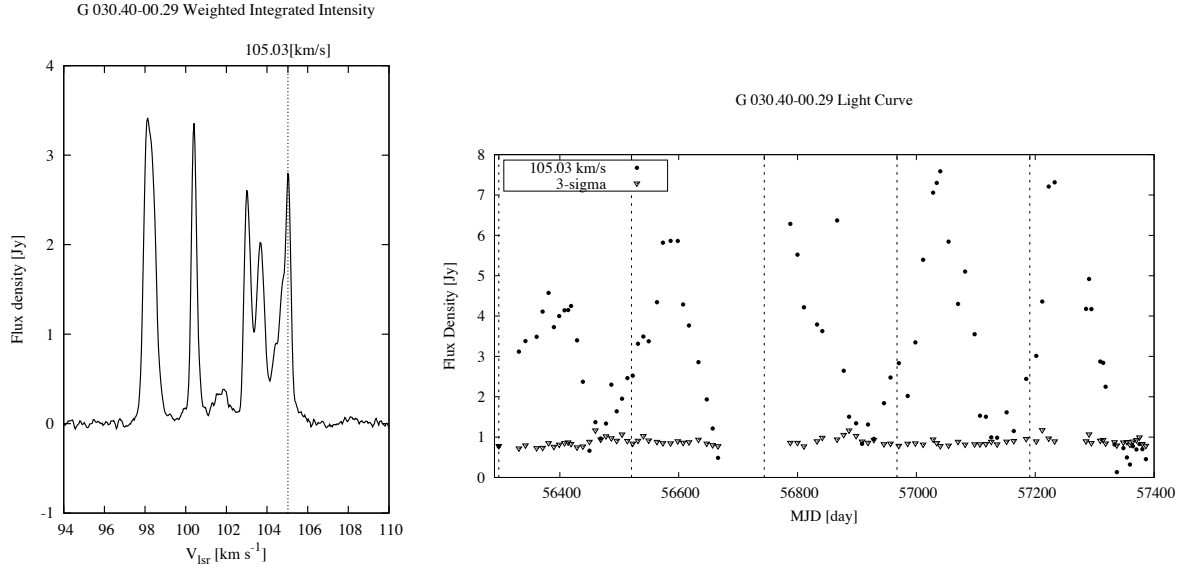


図 25 G 030.40-00.29 の強度変動時系列

図 24 G 030.40-00.29 のライン
プロファイル

表 14 G 033.64-00.21 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
57.04	0.484	1.3	5.61	0.999238	-
58.94	0.395	71.4	72.32	0.0×10^{-19} 以下	189.94(*)
59.3	0.949	91.2	75.93	0.0×10^{-19} 以下	-
59.63	1.458	611.8	23.74	2.3×10^{-7}	-
60.32	0.23	576.9	56.82	0.0×10^{-19} 以下	-
60.93	0.23	232.1	30.4	5.0×10^{-10}	-
62.69	0.201	41.7	37.36	2.1×10^{-13}	256.97
63.16	0.231	40.6	37.23	2.1×10^{-13}	256.97

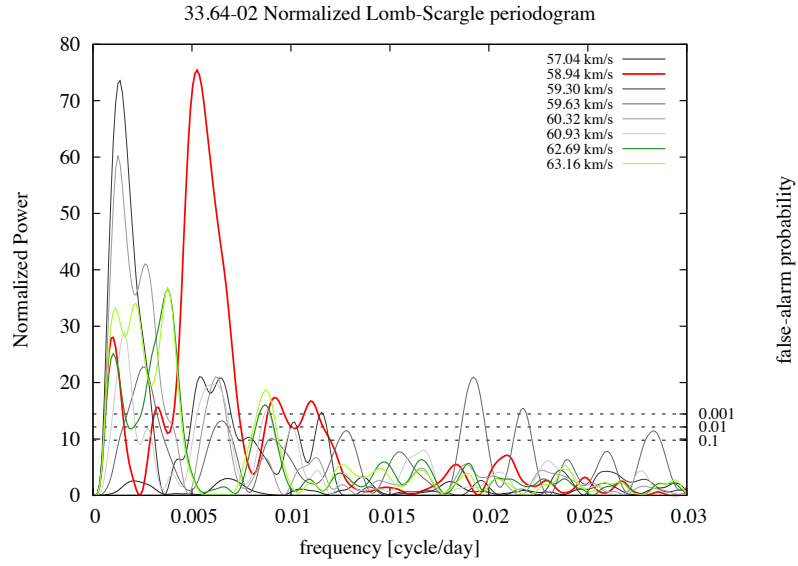


図 26 G 033.64-00.21 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

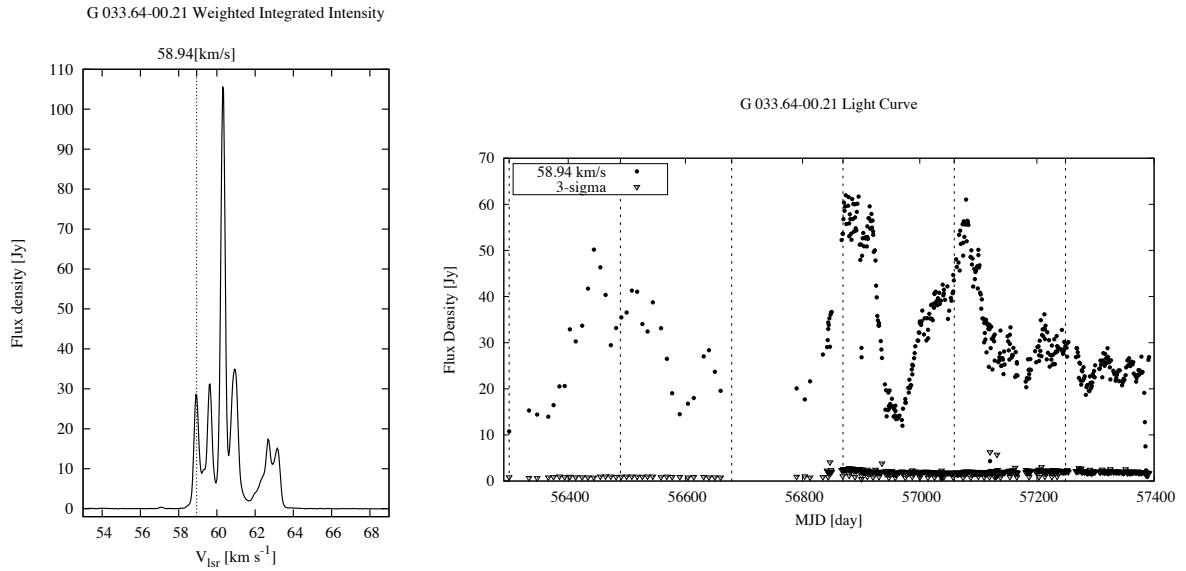


図 28 G 033.64-00.21 の強度変動時系列

図 27 G 033.64-00.21 のライン
プロファイル

4.2.8 G 034.39+00.24

G 034.39+00.24 は、8 個の速度成分が同定されている。複数の速度成分で全ての選出条件を満たすのは、この天体と次に紹介する G 036.70+00.09 だけである。G 034.39+00.24 の速度成分の内、誤警報確率 $\alpha \leq 0.001$ であるのは、選出条件をすべて満たす視線速度 55.67[km/s] と 60.46[km/s] のみである。また、導出される周期は両速度成分とも 29.33 日であり、隣合う独立周期との差は +0.10 日、−0.09 日であり、G 014.23-00.50 に次いで 2 番目に短い。

4.2.9 G 036.70+00.09

G 036.70+00.09 は、7 個の速度成分が同定されている。新検出の 11 天体の内、最も多くの速度成分が全ての選出条件を満たしている。全ての選出条件を満たす速度成分は、全部で 4 つある。4 つの内、視線速度 53.11[km/s], 55.09[km/s], 62.10[km/s] の成分の導出周期は 53.46 日である。残る視線速度 53.65[km/s] の成分の導出周期は 53.14 日であり、他の 3 つの導出周期 53.46 日と隣合う独立周期である。周期 53.46 日の隣合う独立周期との差は +0.33 日、−0.32 日であり、周期 53.14 日の隣合う独立周期との差は ± 0.32 日である。

表 15 G 034.39+00.24 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
55.67	0.337	138.4	17.73	0.000008	29.33(*)
57.9	0.364	7.2	8.17	0.111982	-
59.96	0.492	8.2	9.92	0.020431	-
60.46	0.388	13.4	18.62	0.000003	29.33(*)
60.82	0.356	4.2	7.7	0.172474	-
61.39	0.264	8	9.42	0.033583	-
62.54	0.175	16	9.74	0.024367	-
63.43	0.468	5	8.96	0.052654	-

表 16 G 036.70+00.09 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
52.23	0.55	5.5	4.94	0.949595	-
52.56	0.292	4.5	4.23	0.991436	-
53.11	0.326	13.2	19.82	0.000001	53.46(*)
53.65	0.656	18.6	20.77	0.000001	53.14(*)
55.09	0.579	13.7	28.36	1.9×10^{-10}	53.46(*)
62.1	0.384	22.6	26.35	1.8×10^{-9}	53.46(*)
62.93	0.5	4.5	16.48	0.000028	53.79

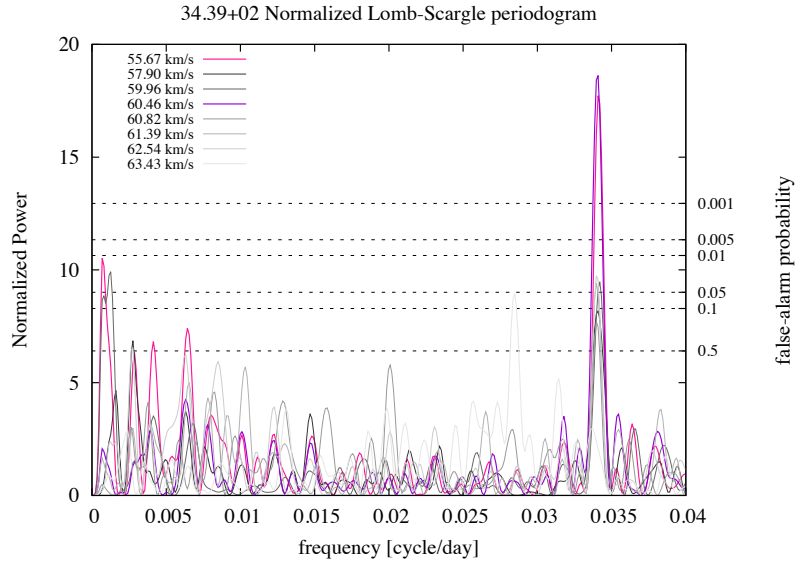


図 29 G 034.39+00.24 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

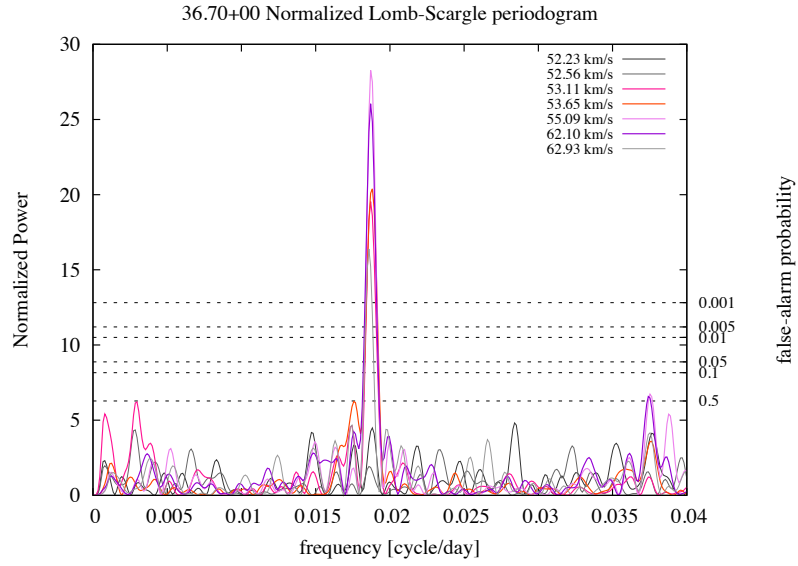


図 30 G 036.70+00.09 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

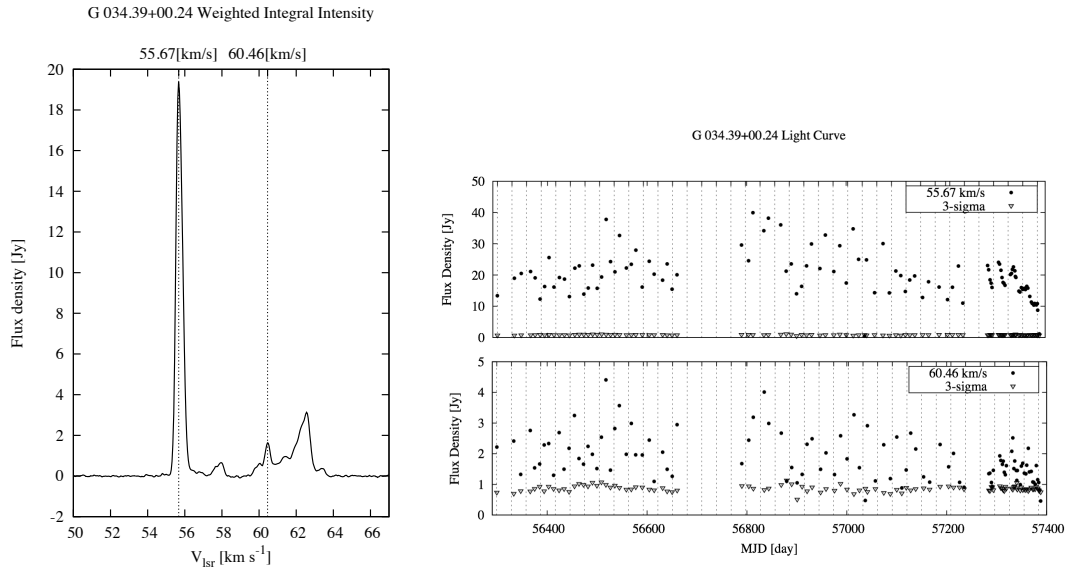


図 31 G 034.39+00.24 のラインプロファイル

図 32 G 034.39+00.24 の強度変動時系列

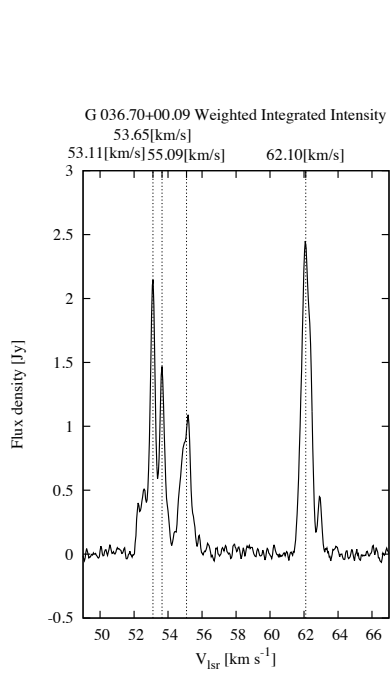


図 33 G 036.70+00.09 のラインプロファイル

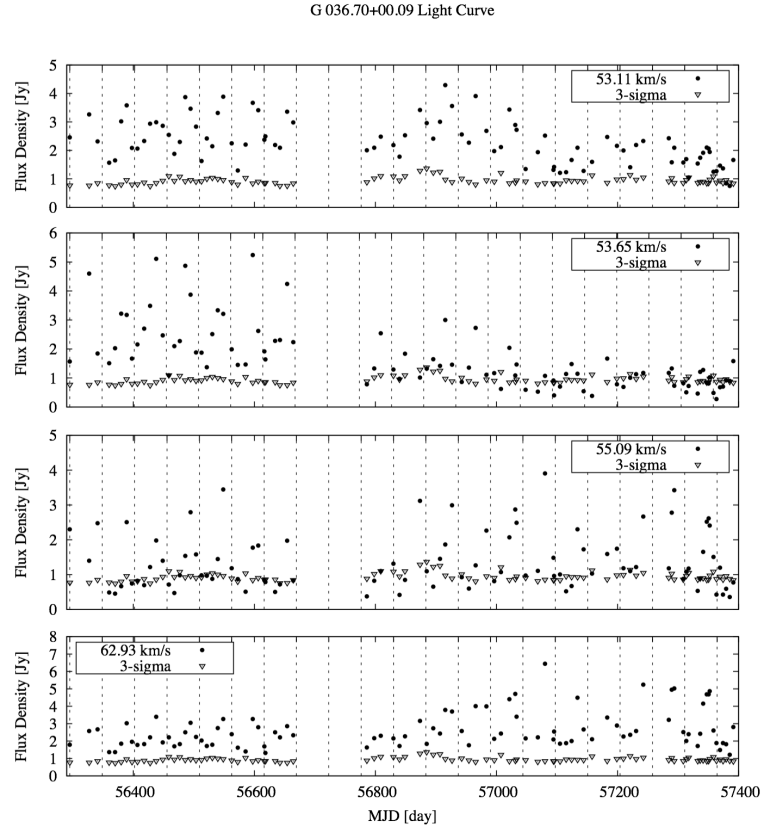


図 34 G 036.70+00.09 の強度変動時系列

4.2.10 G 037.47+00.11

G 037.47+00.11 は、13 個の速度成分が同定されている。この内、選出条件をすべて満たすのは視線速度 54.51[km/s] の成分のみである。他の速度成分は、誤警報確率 α が ≥ 0.001 である。視線速度 54.51[km/s] の成分の導出周期は 68.54 日であり、隣合う独立周期との差は ± 0.54 日である。この視線速度 54.51[km/s] の成分は、選出条件をすべて満たすものの、新検出 11 天体の中で正規化 Lomb-Scargle ピリオドグラムの最大値と誤警報確率 $\alpha = 0.001$ のときの閾値との差が最も小さい。すなわち、最も誤警報確率が大きい結果である。

表 17 G 037.47+00.11 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
49.99	0.235	5.1	3.34	0.999995	-
52.5	0.316	5.5	4.09	0.996823	-
53.97	0.534	56.4	7.36	0.194948	-
54.51	0.554	77	14.4	0.000190	68.54(*)
56.83	0.134	37.6	7.47	0.175808	-
57.17	0.187	15.5	7.96	0.111524	-
58.44	0.212	35.3	11.64	0.003002	-
59.41	0.164	24.1	11.64	0.002995	-
59.99	0.269	17.2	8.88	0.046433	-
61.26	0.198	31.1	12.54	0.001217	-
61.68	0.137	30.2	11.26	0.004352	-
62.08	0.144	27.9	11.63	0.003032	-
62.55	0.122	29.6	12.68	0.001057	-

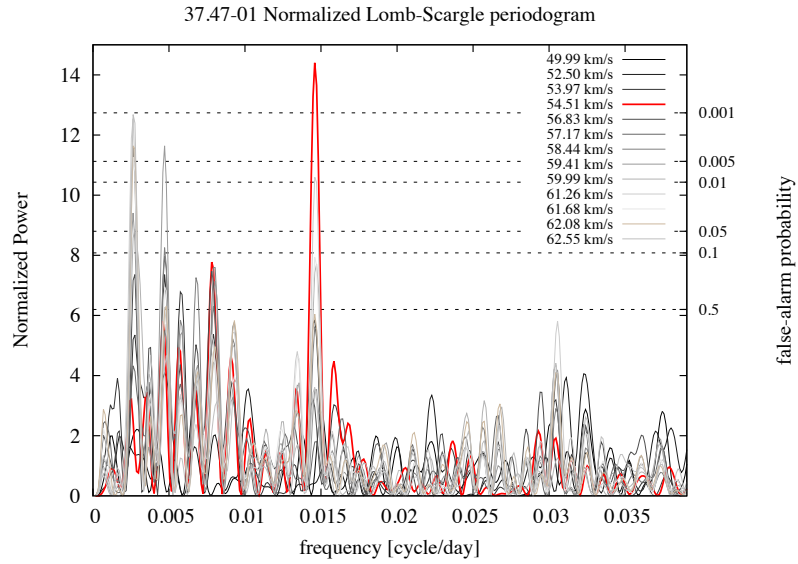


図 35 G 037.47+00.11 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

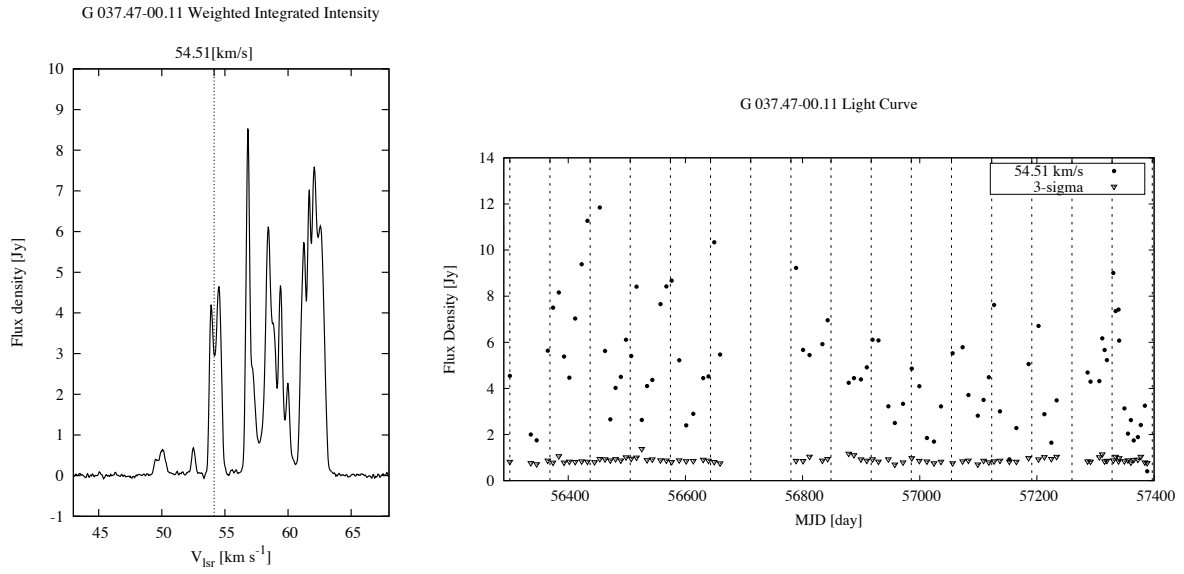


図 37 G 037.47+00.11 の強度変動時系列

図 36 G 037.47+00.11 のライ
ンプロファイル

4.2.11 G 174.19-00.09

G 174.19-00.09 は、6 個の速度成分が同定されている。選出条件をすべて満たすのは、視線速度 2.04[km/s] の成分のみである。視線速度 0.37[km/s] の成分は、最大強度対装置雑音比 $S_{\text{max}}/1\sigma$ が 7 以下のため、選出されなかった。残りの 4 つの速度成分は、導出周期が観測期間 1100 日以内に 3 周期分含まれないため、選出されなかった。選出条件をすべて満たす視線速度 2.04[km/s] の成分の導出周期は 46.81 日であり、隣合う独立周期との差は ± 0.25 日である。

表 18 G 174.19-00.09 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
0.37	0.54	3	20.29	0.000003	314.28
1.48	0.296	187.2	92.35	0.0×10^{-19} 以下	-
2.04	0.455	32.7	43.74	0.0×10^{-19} 以下	46.81(*)
3.77	0.544	179.3	76.74	0.0×10^{-19} 以下	-
4.55	0.369	71.7	71.15	0.0×10^{-19} 以下	-
4.82	0.502	35.1	123.75	0.0×10^{-19} 以下	-

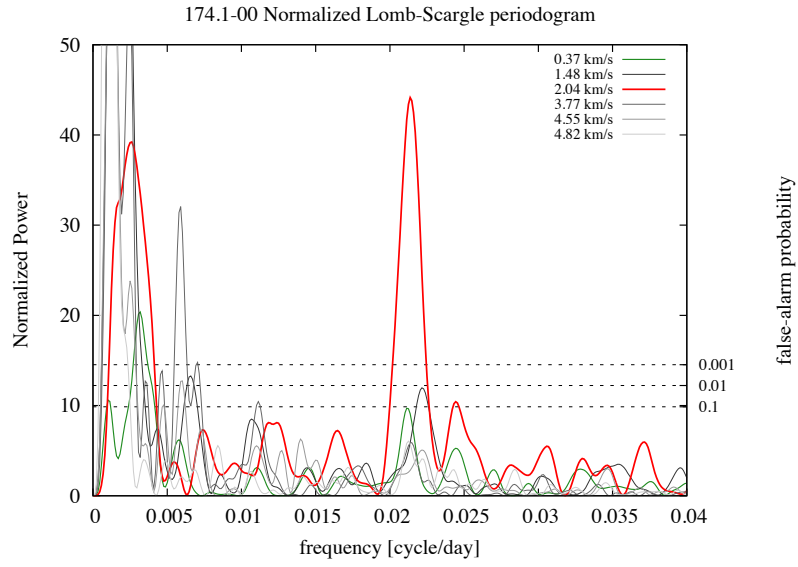


図 38 G 174.19-00.09 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

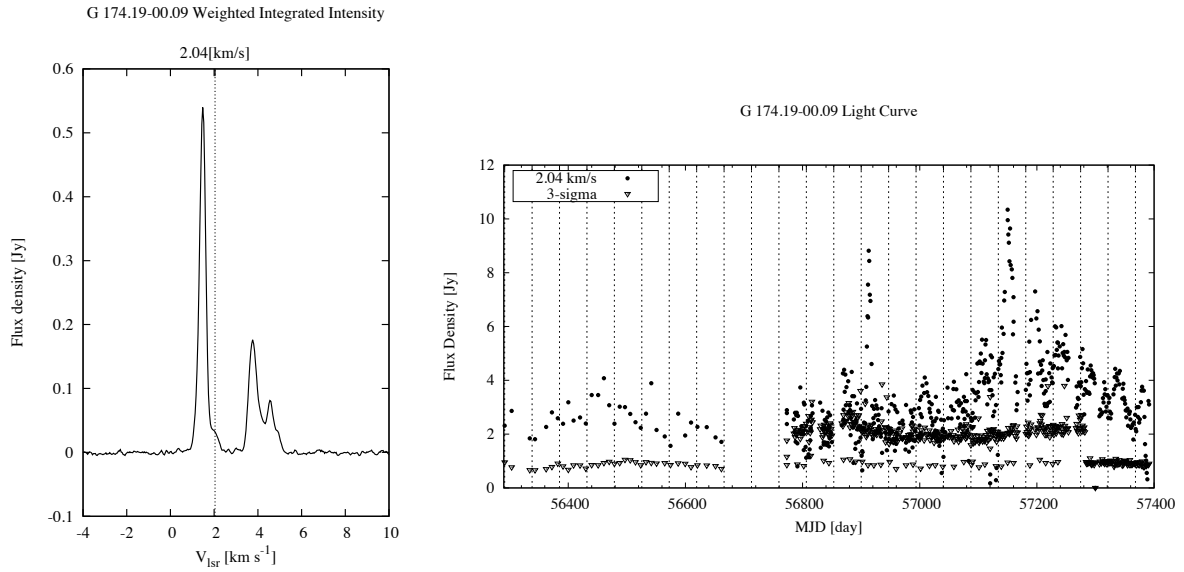


図 40 G 174.19-00.09 の強度変動時系列

図 39 G 174.19-00.09 のライン
プロファイル

4.2.12 G 196.45-01.67

G 196.45-01.67 は, Goedhart et al.(2004) によって, 視線速度 14.73[km/s] の成分に周期 668 日の強度変動があると報告されている。Goedhart et al.(2004) の観測期間は, 1999 年 1 月から 2003 年 3 月であり, 通算日数は約 1500 日である。得られたデータ数は 198 個である。

日立アンテナ・モニター観測では, G 196.45-01.67 は 4 個の速度成分が同定されている。選出条件をすべて満たすのは, 視線速度 14.70[km/s] の成分のみである。Goedhart et al.(2004) によって得られたものと同じ視線速度成分と思われる。導出周期は 146.53 日であり, 隣合う独立周期との差は +2.43 日, -2.40 日である。導出周期 146.53 日は, Goedhart et al.(2004) によって得られた周期 668 日の約 1/5 である。実際, Goedhart et al.(2004) では, G 196.45-01.67 は厳密な周期性を持つと分類しながらも, より長い期間観測されれば厳密な周期性は見つからない可能性もあると示唆している。すなわち, Goedhart et al.(2004) では, 報告周期 668 日を否定はできない結果とし, その理由として報告周期 668 日確かめるのに十分な観測期間が得られていないと考察している。

表 19 G 196.45-01.67 の解析結果

視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
14.70	0.348	120.9	43.83	0.0×10^{-19} 以下	146.53(*)
15.15	0.253	78.7	42.02	0.0×10^{-19} 以下	146.53
15.43	0.289	65.9	45.49	0.0×10^{-19} 以下	145.53
15.88	0.454	79.1	62.52	0.0×10^{-19} 以下	1098.97

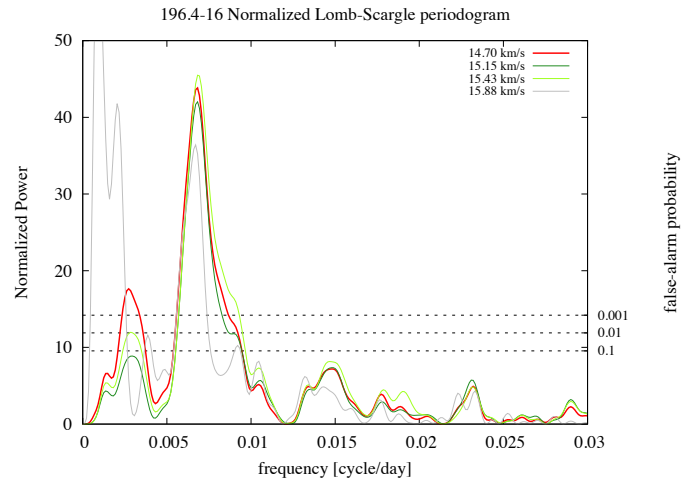


図 41 G 196.45-01.67 の正規化 Lomb-Scargle ピリオドグラムのスペクトル

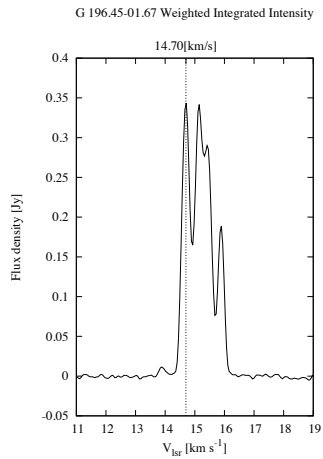


図 42 G 196.45-01.67 のライン
プロフィール

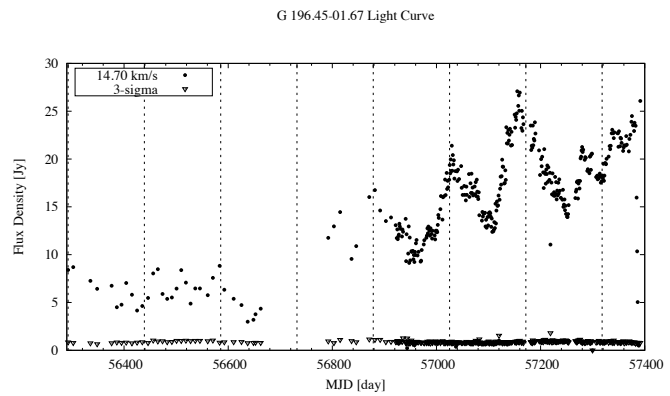


図 43 G 196.45-01.67 の強度変動時系列

表 20 日立アンテナ・モニター観測による結果

天体名	視線速度 V_{lsr} [km/s]	変動係数 I_{mod}	$S_{\text{max}}/1\sigma$	$P'_N(\omega_{\text{max}})$	誤警報確率 α	周期 T [day]
G 045.47+00.13	59.65	0.535	25.7	18.65	0.0000	198.18
	65.10	0.352	24.0	15.55	0.0001	198.18
G 045.49+00.13	59.67	0.569	27.4	19.60	0.0000	197.27
	65.11	0.357	13.5	15.30	0.0001	201.86

表 21 2 天体の位置 (Pandian et al. 2011 参照)

天体名	赤経 (J2000.0) [h m s]	赤緯 (J2000.0) [° ' "]
G 045.47+00.13	19 14 7.362	+11 12 15.98
G 045.49+00.13	19 14 11.357	+11 13 6.41

4.3 選出除外天体

今回、解析した 443 天体の内、これまで紹介した 12 天体に加え、G 045.49+00.13 の 1 天体が選出条件のすべてを満たした。しかし、既知の周期変動天体として Szymczak et al. (2015) によって報告されている G 045.47+00.13 の速度成分が混入した恐れがあると判断し、新検出としなかった。表 20 に日立アンテナ・モニター観測によって得られた G 045.49+00.13 と G 045.47+00.13 の選出条件をすべて満たす速度成分の結果をまとめた。なお、Szymczak et al. (2015) によって報告された G 045.47+00.13 の結果は、視線速度 59.60[km/s] の成分で周期 195.6 日である。日立アンテナ・モニター観測の結果である、視線速度 59.65[km/s] の成分で周期 198.18 日との差は 2.58 日なので、ほぼ等しいと言える。日立アンテナ・モニター観測では、視線速度 65.10[km/s] の成分も選出条件を全て満たしているため、新たな速度成分で新検出されたと主張したいところである。実は、この速度成分も、Szymczak et al. (2015) で周期性が報告されている。しかし、彼らは、周期変動を示す視線速度 59.60[km/s] の成分が同期した影響があるとして、視線速度 64.2-66.3[km/s] の成分を除外している。それに倣い、日立アンテナ・モニター観測の結果からも視線速度 65.10[km/s] の成分は除外することにする。一方、G 045.47+00.13 とは別に日立アンテナ・モニター観測によって得られた G 045.49+00.13 の結果は、選出条件を全て満たすため、新検出の周期変動天体として扱いたいところである。まず、これらの天体について、日立アンテナ・モニター観測によって得られた G 045.49+00.13 の速度成分と導出周期が、同観測によって得られた G 045.47+00.13 とほぼ等しいことがわかる。日立アンテナ・モニター観測によって得られた G 045.49+00.13 の観測結果には、視線速度 65.11[km/s] の成分が含まれているが、Pandian et al. (2011) による 6.7GHz

メタノールメーザーの高い空間分解能 ($\sim 50[\text{mas}]^{*9}$) の観測によると、G 045.49+00.13 には約 65[km/s] の速度成分は観測されていない。また、彼らの観測によると G 045.47+00.13 には、視線速度 59[km/s] 付近と 65[km/s] 付近と 62[km/s] 付近の速度成分が確認されたと報告している。次に、表 21 に示した 2 天体の赤経・赤緯から 2 天体の離角は 77.4 秒角であり、離角は小さいことがわかる。日立アンテナの空間分解能は約 4.6 分角なので、この 2 天体を切り分けることは困難であると思われる。この 2 つの理由から、G 045.49+00.13 は G 045.47+00.13 の速度成分が混入した可能性が高いため新検出の周期強度変動天体と見なしていない。

*9 mas; milliarcsec; ミリ秒角

4.4 考察

4.4.1 選出条件の考察

§4.1 で説明した 4 つの選出条件は、比較的厳しい条件を課していることになる。もちろん、本研究で取り扱う雑音の定義が大雑把なことを考慮するためである。しかし、選出条件によっては新検出と主張される周期変動天体は一部で変わる可能性がある。

まず、一般に用いられる仮説検定では、誤警報確率 α は、0.05 もしくは 0.01 を閾値として設定することが多い。これに比べ、本研究では、誤警報確率 $\alpha = 0.001$ を閾値としているため、およそ 10 倍厳しい条件を課していることになる。次に、周期変動していない成分の影響を考慮するため、導出周期の 3 周期分が観測期間に収まる条件を課した。この条件は、導出周期の再現性を計るためにも有効である。また、図 10 のように、天体の強度変動図に導出周期ごとの格子線を挿入することによって導出周期の再現性を計ることも可能である。ただし、この場合、目視による判断を行うため本研究の目的に反することになる。目視による判断を許すならば、強度変動図に格子線を挿入する他に、導出周波数を持つ正弦波を強度変動図に挿入する方法 (e.g., Szymzak et al. 2015; Goedhart et al. 2007) もある。次に、検出力の確保と独立周波数の分解能を考慮した条件 3. について考察する。確かに、図 7 からわかる通り、独立周波数の数 N の増加に伴い、検出力が小さくなる。ただし、付録 E 表 21 に載せる解析対象天体のデータ数は、近いうちにほとんどがデータ数 $N_0 \sim 100$ になる。すなわち、独立周波数の数は $N \sim 50$ であり、オーバーサンプリングにより独立周波数の数が増えても検出力は少ししか減少しないことが言える。しかし、§2.4 で定義した独立周波数の数 $N = N_0/2$ が、少なく見積もられている影響を考慮すべきでないかという意見もあるだろう。つまり、多くの研究で Lomb-Scargle 法で求まる独立周波数の数は、 $N \geq N_0/2$ と主張されているのに、本研究で $N = N_0/2$ と定義するのは、検出力を大きく見積もっているのではないかという意見である。この点に関して、次のように考えている。Scargle(1982) では、誤警報確率 α を $\ll 1$ とした場合の閾値 z_0 (式 (12)) は、

$$\begin{aligned} z_0 &= -\ln \left[1 - (1 - \alpha)^{1/N} \right] \\ &\approx \ln(N/\alpha) \end{aligned}$$

で近似できると示している。したがって、一般的に用いられる誤警報確率より小さい値を閾値として用いるのであれば、 N の見積もりはより一層、正確である必要はない。実際、Numerical Recipes in C(Press et al. 1988) では、正規化 Lomb-Scargle ピリオドグラムの最大値の誤警報確率を導出する際に、 $\alpha \approx N \exp(-z_0)$ を用いている。最後に、電波天文観測の特徴と強度変動を示さない天体、すなわち $\omega = 0$ の直流成分を切り離すために条件 4. を課した。初めにこの条件を課すと、観測対象の約 450 天体の内、条件を満たす天体は約 150 天体しかないと報告されている(茨城大学杉山孝一郎氏)。しかし、変動係数、及び、最大強度対装置雑音比の基準値は任意であるため、今回の条件 $I_{\text{mod}} \geq 0.3$, $S_{\text{max}}/1\sigma \geq 7$ を変更することも可能である。なお、天文学的な要素を考慮した条件 4. を課さない場合、すなわち得られたデータを絶対的な時系列データとして取り

扱う場合、条件 1.~3. を満たす天体は 35 個である。以上のように、選出条件は一意的に定義できるものではないと言える。

4.4.2 既知の周期変動天体: 日立アンテナ・モニター観測

選出条件は今回設定したものを用いて、帰納的に本研究で用いた Lomb-Scargle 法の妥当性を考察する。すなわち、既知の周期強度変動天体でモニター観測の対象天体の内、何天体が選出条件をすべて満たしたのかを見ていく。既知の周期強度変動天体の内、日立アンテナ・モニター観測の対象天体は 12 天体である。表 1 に示した既知の周期強度変動天体を日立アンテナでモニター観測したデータの内、全ての選出条件を満たしたものを表 22 に示す。表 22 中の先行報告の周期とは、表 1 に示した参考文献にて報告されている周期である。ただし、G 196.45+01.67 は §4.2.12 で紹介した通り、先行報告とは大きく異なる周期が導出されたので除いている。表 22 に示した天体は、G 037.55+00.20 を除く全ての天体で先行報告と導出周期の差が数日以内であることがわかる。ここで、G 037.55+00.20 は、先行報告の Araya et al. (2010) にて、厳密な周期性があるとは主張していない。Araya et al. (2010) の表現を引用すると、G 037.55+00.20 は「quasi-periodic」*¹⁰ である。このことから、G 037.55+00.20 の先行報告と導出周期の差が 12.37 日とやや大きくなってしまうことは考えられる。しかし、モニター観測対象天体の内、残る 2 天体 (G 012.68-00.18, G 188.94+00.88) は選出条件を満たさなかった。それぞれの先行報告の周期は、307 日と 416 日である。後者、G 188.94+00.88 の先行報告周期 416 日は、その 3 周期分 (1248 日) が観測期間を上回ってしまうため選出されなかったと思われる。一方、G 012.68-00.18 は、先行報告の Goedhart et al. (2004) において「quasi-periodic」と結論付けており、周期は 307 ± 60 日であると報告している。日立アンテナ・モニター観測のデータにおいても、Lomb-Scargle 法の解析結果が誤警報確率 0.001 以下である速度成分の導出周期は 1245.73 日となっており、非常に疑わしい結果が得られている。すなわち、非周期的な強度変動成分が含まれる可能性がある。

いずれにせよ、厳密な周期性があると報告されている周期強度変動天体に対して先行報告と導出周期の差が数日以内であることから、本研究で用いた Lomb-Scargle 法の妥当性がわかる。同時に、日立アンテナ・モニター観測の有益性も確認できる。

*¹⁰ quasi- : 「準」を意味する。すなわち、準周期的という主張である。

表 22 既知の周期変動天体: 日立アンテナ・モニター観測の結果

天体名	先行報告の周期 [day]	導出周期 [day]	差の絶対値 [day](変化率 %)
G 009.62+00.19	243	244.21	1.21 (+0.50%)
G 012.88+00.48	29.5	29.53	0.03 (+0.10%)
G 022.36+00.06	179.3	175.51	3.79 (−2.11%)
G 025.41+00.10	244.7	250.51	5.81 (+2.37%)
G 037.55+00.20	237	249.37	12.37 (+5.22%)
G 045.47+00.13	195.6	198.18	2.58 (+1.32%)
G 073.06+01.80	159	157.13	1.87 (−1.18%)
G 075.76+00.34	119.9	118.91	0.99 (−0.83%)
IRS22198+6336	34.6	34.38	0.22 (−0.64%)

4.4.3 原始星の脈動周期と光度関係

最後に、茨城大学 杉山孝一郎氏によって報告された、新検出の周期変動天体を含めた原始星の脈動周期と光度関係を図 44 に示す^{*11}。図 2 同様に、式 (1) で表される線を破線で表している。杉山氏の報告によると、式 (1) の関係は、

$$\log \left(\frac{L_*}{L_{\odot}} \right) = 4.40(\pm 0.14) + 0.47(\pm 0.41) \log \left(\frac{P_*}{100[\text{days}]} \right)$$

で改められる可能性を指摘している。今後の展開に関しては、杉山孝一郎氏の研究を追っていただきたい。

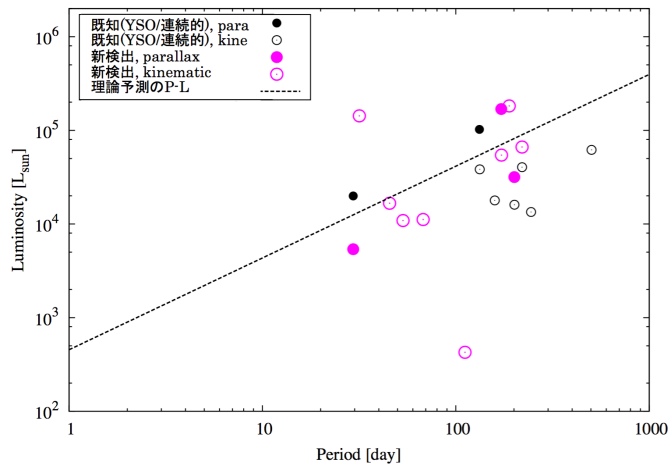


図 44 図 2 に新検出の周期強度変動天体を加えたもの。

^{*11} 選出条件 3. を課していないため、G 033.40+00.01(31.7 日), G 028.02-00.44(111.5 日), G 111.53+00.76(171.9) が含まれている (() 内は周期日数)。また、変動傾向が連続的であるものに注目しているため、変動傾向が間欠的であると判断された G 014.23-00.50 は除かれている。

5 まとめ

任意の観測日間隔で得られた時系列データに対して、そのパワースペクトルの推定に当たるピリオドグラムを Lomb-Scargle 法を用いて導出した。同時に、求まる最大値の有意性を評価するために、帰無仮説「観測値は正規分布する雑音のみである」の誤警報確率を求めた。4つの選出条件を課し、正規化 Lomb-Scargle ピリオドグラムが最大値をとる周波数の逆数を周期強度変動の周期 [日] として導出した。日立アンテナでモニター観測された 6.7GHz メタノールメーザーの時系列データを解析し、

- 既知の周期的な強度変動を示す 6.7GHz メタノールメーザー 17 天体の内、観測を行っている 12 天体中 9 天体で同様の周期を確認
- 既知の周期的な強度変動を示す 6.7GHz メタノールメーザー 17 天体の内、1 天体で先行報告とは大きく異なる周期を導出
- 新たに周期的な強度変動を示す 6.7GHz メタノールメーザー 11 天体を選出

という結果が得られた。

付録 A メーザー

メーザーとは、粒子のエネルギー準位が反転分布している状態から、誘導放射によって低い準位へカスケード的に遷移することによって起こる強い放射（輝線）のことである。ガス密度が高いところでは、粒子同士の衝突が支配的、すなわち熱平衡状態にあるため、粒子のエネルギー準位はボルツマン分布で与えられる。ボルツマン分布は、2つの準位を考えると、粒子数 n_1, n_2 、統計的重みを g_1, g_2 として、

$$\frac{n_2}{n_1} = \frac{g_2}{g_1} \exp\left(-\frac{h\nu}{kT}\right) < 1$$

で与えられる。ここで、 h はプランク定数、 ν は周波数、 k はボルツマン定数、 T は励起温度である。しかし、ガス密度が希薄な宇宙空間では、粒子同士の衝突よりも自発放射や誘導放射が支配的になる場合がある。このとき、統計的重み当たりの粒子数の関係は、

$$\frac{n_1}{g_1} < \frac{n_2}{g_2}$$

となる。この状態で、例えば、2準位間にエネルギー $h\nu_{21}$ を持つ電磁波が入射してくると、上の準位から下の準位へ粒子が落ちて周波数 ν_{21} の電波を誘導放射する。この放射が、メーザー放射である。

付録 B 定常過程とエルゴート過程

時系列データ $X_i = X(t_i)$, $i = 1, 2, \dots, N_0$ の期待値

$$\mu_i = E(X_i)$$

を時系列 X_i の平均値関数とする。また、時間を k だけシフトした X_{i-k} との共分散

$$\text{Cov}(X_i, X_{i-k}) = E[(X_i - \mu_i)(X_{i-k} - \mu_{i-k})]$$

を時系列 X_i の自己共分散関数という。 $k = 0$ のとき、

$$\text{Var}(X_i) = E[(X_i - \mu_i)^2]$$

を時系列 X_i の分散関数という。

時間を l だけシフトしたとき、これらの値が

$$E(X_i) = E(X_{i-l}) = \mu$$

$$\text{Cov}(X_i, X_{i-k}) = \text{Cov}(X_{i-l}, X_{(i-k)-l}) = C_k$$

$$\text{Var}(X_i) = \text{Var}(X_{i-l}) = C_0$$

と変化しない場合、時系列 X_i には定常性がある、または定常過程という。すなわち、時系列データの確率密度関数が時間シフトに対して変化しないことを指す。簡単に言えば、時系列の統計的性質が時不変であることを意味する。

時系列 $X(t_i)$ の集合 $\{X(t_i)\}$, $i = 1, \dots, N_0$ から標本 $X_k(t_i)$ を抽出したときの時間平均は、

$$X_k = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} X_k(t_i) dt$$

で定義される。観測データに対する時間平均は、全観測期間などの時間帯 T における強度の平均にあたる。

ある時刻 t の時系列の集合 $\{X(t)\}$ を考え、集合全体で平均をとった

$$\langle X(t) \rangle = \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K X_k(t)$$

を集合平均 (アンサンブル平均) という。ここで K は、本研究では観測例の数にあたるが、観測結果は時刻 t について、一つの観測例しか与えられていない。すなわち、 $K = 1$ である。

時系列 $X(t_i)$ が定常過程であり、ある標本についての時間平均と集合平均が等しいとき、

$$X_k = \langle X(t) \rangle,$$

時系列 $X(t_i)$ はエルゴート性を持つという。エルゴート過程であれば定常過程であるが、定常過程であってもエルゴート過程とは限らない。本研究で取り扱う、(正規化)Lomb-Scargle ピリオドグラムは、定常性とエルゴート性を持つ時系列に対してのみ有効である。

付録 C Lomb-Scargle ピリオドグラムの時不変性

式 (4) で表されるピリオドグラムのすべての j に対して、 $t_j \rightarrow t_j + T_0$ となる時間シフトを考えると、

$$\begin{aligned} P(\omega) &= \frac{1}{N_0} \left| \sum_{j=1}^{N_0} X(t_j + T_0) \exp[-i\omega(t_j + T_0)] \right|^2 \\ &= \frac{1}{N_0} \left\{ \left| \sum_{j=1}^{N_0} X(t_j + T_0) \exp(-i\omega t_j) \exp(-i\omega T_0) \right| \times \left| \sum_{j=1}^{N_0} X(t_j + T_0) \exp(i\omega t_j) \exp(i\omega T_0) \right| \right\} \\ &= \frac{1}{N_0} \left[\left(\sum_{j=1}^{N_0} X(t_j + T_0) \cos \omega(t_j + T_0) \right)^2 + \left(\sum_{j=1}^{N_0} X(t_j + T_0) \sin \omega(t_j + T_0) \right)^2 \right] \end{aligned}$$

となり、時間シフトに対して不変である。すなわち、式 (4) のピリオドグラムは時不変である。特に、位相 $\exp(-i\omega T_0)$ とその複素共役の積は 1 であることに由来する。一方、 τ を加える前の

Lomb-Scargle ピリオドグラム (式 (6)),

$$P_X(\omega) = \frac{1}{2} \left[\frac{\left(\sum_{j=1}^{N_0} X(t_j) \cos \omega t_j \right)^2}{\sum_{j=1}^{N_0} \cos^2 \omega t_j} + \frac{\left(\sum_{j=1}^{N_0} X(t_j) \sin \omega t_j \right)^2}{\sum_{j=1}^{N_0} \sin^2 \omega t_j} \right] \quad (18)$$

は正規分布に従う雑音に対して指数分布を持つが, $t_j \rightarrow t_j + T_0$ の時間シフトに対して時不変ではない。一般に, ピリオドグラムが推定しているパワースペクトルは, 位相に関わりなく信号に含まれる調和関数要素を計算する意味を持つ。したがって, 時間シフトした時系列との間に位相遅れ (位相ずれ) がある場合, 位相遅れをなくすように修正を加える必要がある。時間シフト前の $X(t_j)$ と時間シフトした $X(t_j + T_0)$ の離散フーリエ変換後の位相差 θ_{lag} は,

$$\theta_{\text{lag}} = \frac{1}{2} \tan^{-1} \left(\frac{\sum_{j=1}^{N_0} \sin 2\omega t_j}{\sum_{j=1}^{N_0} \cos 2\omega t_j} \right)$$

で表される^{*12}。この θ_{lag} を周波数 ω で割った,

$$\tau = \frac{\theta_{\text{lag}}}{\omega} = \frac{1}{2\omega} \tan^{-1} \left(\frac{\sum_{j=1}^{N_0} \sin 2\omega t_j}{\sum_{j=1}^{N_0} \cos 2\omega t_j} \right)$$

は時間シフト前後の時系列の位相遅れを周波数領域で表すことができる。式 (18) の時間を表す変数から τ を引くことにより, ピリオドグラムの位相遅れをなくすることができる。

付録 D プログラム lsperiodogram

ここでは, 日立アンテナ・モニター観測によって得られたデータを対象に正規化 Lomb-Scargle ピリオドグラムを計算する C 言語プログラミングを掲載する。正規化 Lomb-Scargle ピリオドグラムの計算に当たるアルゴリズムは, Numerical Recipes in C (Press et al. 1988) の 13.8 Spectral Analysis of Unevenly Sampled Data を参考にしている。その他の処理や計算は, C 言語プログラムの特徴である関数を用いたりしている。

まず, 日立アンテナ・モニター観測のデータの記録書式を説明する。モニター観測のデータは, 最終的にファイル名「天体名 (半角 8 文字)_rcparam.dat」(e.g., 4.434+01_rcparam.dat) に記録される。ファイルの中身は,

```
#source code : (天体名) ←
#year MJD 1sigma integ.int.(Jy km/s) Vlsr(km/s) peak(Jy) S/N ...←
(日付) (修正ユリウス日) (1σ) (積分強度 [Jy km/s]) 1 成分目の (視線速度 [km/s]) (ピーク強度 [Jy]) (ピーク強度 1
σ 比) 2 成分目...n 成分目 ←
```

の形式で書かれる。1 列目の日付は西暦を有効数字 4 桁で表したものであり, 2 列目にその修正ユリウス日 (MJD) を表している。3 列目の 1σ は, §4.1 節で与えた全観測値の標準偏差である。4 列目の積分強度とは, 観測されたフラックス密度を各天体で指定した視線速度範囲にわたって積分

^{*12} 詳細な導出には, 両者のクロススペクトルを考えていくことになる

した値であり、フラックス密度-視線速度スペクトル (e.g., 図 39) の面積にあたる。5 列目から 7 列目は、1 成分目の視線速度、強度の最大値、最大強度を 1σ で割った値がそれぞれ記録される。8 列目から 10 列目は、2 成分目の各値であり、11 列目からも 3 列ごとに成分数まで与えられる。なお、 $\leftarrow$ は改行を表す。プログラムでは、「(天体名)_rcparam.dat」の中身を一度、2 次元配列 rcp[][] に格納する。その後、2 列目 (rcp[i][1]) の修正ユリウス日を t_i とし、そのときの各成分の強度の最大値 (6 列目, 9 列目, …(rcp[i][5], rcp[i][8], …)) を式 (2) の $X(t_i)$ として与える。それぞれ、関数 period の引数 x[], y[] に相当する。

D.1 プログラム lsperiodogram のソースコード

それでは、プログラム lsperiodogram を掲載する。

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <math.h>
5 #define SIZE 1028
6 #define SIZE2 SIZE*4 //nout
7
8 void avevar(double data[], unsigned long n, double *ave, double *var);
9 void period(double x[], double y[], int n, double ofac, double hifac, double px
    [], double py[], int np, int *nout, int *jmax, double *prob, double *sigma2,
    double *pdc, double *aver);
10 void count(char *filename, int *cc, int *cr, int *skip);
11 void fit(double tofitx[], double tofity[], int cr, double *m, double *l);
12 void significantlevel(double nout, double fap[6]);
13 void gnufile(double fap[6], double pxmax, double pymax, int seibun, char *
    filename, char *result, char *gpname, double tognuv[]);
14 void countname(int *cn);
15
16 int main(int argc, char *argv[]){
17     FILE *fp, *fout2, *fname, *fout3, *fout4;
18     int i=0, j=0, k, p=0, seibun, np = SIZE2, nout, jmax, cc, cr, skip,
        length, namei, cn, q;
19     double m, l, avelo, var_dummy, ave, var, prob, effm, pxmax, pymax, ofac,
        hifac, mjdmax, cycle;
20     char filename[64], result[64], name[64], gpname[64], loadfile[600][64],
        buf[512];
21     double rcp[SIZE][64], tofitmjd[SIZE], tofitx[SIZE], tofity[SIZE], toave[
        SIZE], fluxfit[SIZE], toperiod[SIZE], px[SIZE2], py[SIZE2], fap[6],
        kekka[SIZE2][32], pjmax[30], pro[30], pperiod[30], tognuv[30], pron1
        [30];
22     int jmaximum[30], pickup[30], judge[30];
23
24     double variance[30], average[30];
25     double sigma2, pdc, aver;
26
27     //////////////////////////////////arg////////////////////////////////
28     if (argc != 3 && argc != 4){

```

```

29         ofac = 1;
30         hifac = 1;
31         printf("NOW:ofac=1.0_hifac=1.0[default]\n");
32     }
33
34     if (argc == 1){
35         printf("Analyzed_\\"filenamelist.txt\\"\\n");
36     }
37
38     if (argc == 2){
39         sscanf(argv[1], "%s", filename);
40         strcpy(filename, argv[1]);
41         goto point;
42     }
43
44     if (argc == 3){
45         sscanf(argv[1], "%lf", &ofac);
46         sscanf(argv[2], "%lf", &hifac);
47         printf("NOW:ofac=%3.2lf_hifac=%3.2lf\\n", ofac, hifac);
48     }
49
50     if (argc == 4){
51         fp = fopen(argv[1], "r");
52         sscanf(argv[2], "%lf", &ofac);
53         sscanf(argv[3], "%lf", &hifac);
54         printf("NOW:ofac=%3.2lf_hifac=%3.2lf\\n", ofac, hifac);
55         strcpy(filename, argv[1]);
56         goto point;
57     }
58     ///////////////////////////////////
59     if((fname = fopen("filenamelist.txt", "r")) == NULL){// "filenamelist.txt"
        がない時は
60         printf("filename=");// ファイル名を手入力
61         scanf("%s", filename);
62         strcpy(filename, argv[1]);
63         goto point;
64     }
65
66     countname(&cn);
67     printf("----OPENED_FILENAME(%d)----\\n", cn);
68     for(namei=0; namei<cn; namei++){
69         fgets(filename, sizeof(filename), fname);
70         strtok(filename, "\\n");
71         printf("%s\\n", filename);
72     }
73     point:
74     fp = fopen(filename, "r");
75     length = strlen(filename);
76     strncpy(result, filename, length-12);// _rcparam.dat
77     result[length-12] = '\\0';
78     strcpy(name, result);
79     strcat(result, "_lsperiodogram_result.txt");
80     fout2 = fopen(result, "w");

```

```

81     if(fp == NULL || fout2 == NULL){
82         printf("FILE_OPEN_ERROR\n");
83         exit(1);
84     }
85     //////////count(column*row*#)//////////
86     count(filename, &cc, &cr, &skip);
87     //////////rcparam -> array//////////
88     for(i=0; i<skip; i++){
89         fgets(buf, sizeof(buf), fp);
90     }
91
92     for(i=0; i<cr; i++){
93         for(j=0; j<cc; j++){
94             fscanf(fp, "%lf", &rcp[i][j]);
95         }
96     }
97
98     seibun = (cc-4)/3;
99     j = 0; //initialize
100
101     //////////LS periodogram loop//////////
102     for(k=0; k<cc; k=k+3){
103         if(k+4 >= cc)break;
104         for(i=0; i<cr; i++){
105             toave[i] = rcp[i][k+4]; //column of Vlsr
106             tofity[i] = rcp[i][k+5]; //column of Flux
107             tofitmjd[i] = rcp[i][1]; //column of MJD
108             tofitx[i] = tofitmjd[i] - tofitmjd[0]; //MJD[0]->zero
109         }
110         mjdmax = tofitx[cr-1]; //Obs span
111         //////////Cal. average of Vlsr//////////
112         avevar(toave, cr, &avelo, &var_dummy);
113         tognuv[j] = avelo;
114         //////////fit//////////
115         fit(tofitx, tofity, cr, &m, &l);
116         //////////->toperiod[]//////////
117         for(i=0; i<cr; i++){
118             fluxfit[i] = m * tofitx[i]; //fit function
119             toperiod[i] = tofity[i] - fluxfit[i]; //trend remove
120         }
121         //////////LS algorithm -> array for result//////////
122         period(tofitx, toperiod, cr, ofac, hifac, px, py, np, &nout, &jmax,
123             , &prob, &sigma2, &pdc, &aver);
124
125         jmaximum[j] = jmax;
126         pjmax[j] = py[jmax];
127         pperiod[j] = 1 / px[jmax];
128         pxmax = px[nout-1];
129         pro[j] = prob;
130         cycle = pperiod[j];
131         variance[j] = sigma2;
132         average[j] = aver;

```

```

133          //////////////////////////////////judge////////////////////////////////
134          if(prob<=0.0010)pickup[j]=1;//fap<0.001->"1"
135          else{
136              pickup[j] = 0;
137          }
138          if(prob<=0.0010 && 3*cycle < mjdmax)judge[j]=1;//include 3cycles
              ->"1"
139          else{
140              judge[j]=0;
141          }
142          //////////////////////////////////
143
144          for(p=0; p<nout; p++){
145              kekka[p][0] = px[p];
146              kekka[p][j+1] = py[p];
147          }
148          kekka[nout][j+1] = pdc;
149
150          j++;
151      }
152      //////////////////////////////////LS periodogram loop END////////////////////////////////
153
154      significantlevel(nout, fap);//Cal. false alarm prob.
155
156      pymax = pjmax[0];
157      for (j=0; j<seibun+1; j++) {
158          if (pjmax[j] > pymax) pymax = pjmax[j];
159      }
160      //////////////////////////////////result array->text file////////////////////////////////
161      fprintf(fout2, "#ofac=%3.2lf_hifac=%3.2lf\n", ofac, hifac);
162      fprintf(fout2, "#y(0.001)=_%1f_", fap[0]);
163      fprintf(fout2, "y(0.005)=_%1f_", fap[1]);
164      fprintf(fout2, "y(0.01)=_%1f_", fap[2]);
165      fprintf(fout2, "y(0.05)=_%1f_", fap[3]);
166      fprintf(fout2, "y(0.1)=_%1f_", fap[4]);
167      fprintf(fout2, "y(0.5)=_%1f\n", fap[5]);
168
169      fprintf(fout2, "#nout=_%d\n", nout);//The number of independent freq.
170
171      for(j=0; j<seibun; j++){
172          fprintf(fout2, "#%d_%d", judge[j], pickup[j]);
173          fprintf(fout2, "jmax%2d=_%3d", j+1, jmaximum[j]);
174          fprintf(fout2, "var%2d=_%7.3lf", j+1, variance[j]);
175          fprintf(fout2, "ave%2d=_%7.3lf", j+1, average[j]);
176          fprintf(fout2, "pjmax%2d=_%9.6lf", j+1, pjmax[j]);
177          fprintf(fout2, "fap%2d=_%100.99lf", j+1, pro[j]);
178          fprintf(fout2, "period%2d=_%lf\n", j+1, pperiod[j]);
179      }
180
181      fprintf(fout2, "#frequency_period_power1_power2_...\n");
182
183      fprintf(fout2, "0_0");
184      for(j=0; j<seibun; j++){

```

```

185         fprintf(fout2, "%lf", kekka[nout][j+1]);
186     }
187     fprintf(fout2, "\n");
188     for(p=0; p<nout; p++){
189         fprintf(fout2, "%lf", kekka[p][0]);
190         fprintf(fout2, "%lf", 1/kekka[p][0]);
191         for(j=0; j<seibun; j++){
192             fprintf(fout2, "%lf", kekka[p][j+1]);
193         }
194         fprintf(fout2, "\n");
195     }
196
197     gnuplot(fap, pxmax, pymax, seibun, name, result, gpname, tognuv);
198     strcpy(loadfile[q], gpname);
199     q++;
200
201     fclose(fp);
202     fclose(fout2);
203 }
204 //////////////////////////////////////////////////gnuplot_load_file////////////////////////////////////
205 fout3 = fopen("periodogram.gp", "w");
206 fprintf(fout3, "reset\n");
207 for(q=0; q<cn; q++){
208     fprintf(fout3, "load \"%s\"\n", loadfile[q]);
209 }
210 fprintf(fout3, "reset\n");
211 //////////////////////////////////////
212
213 if(fname != NULL)fclose(fname);
214 fclose(fout3);
215 return 0;
216 }
217
218
219 void period(double x[], double y[], int n, double ofac, double hifac, double px
[], double py[], int np, int *nout, int *jmax, double *prob, double *sigma2,
double *pdc, double *aver){
220     int i, j;
221     double ave, c, cc, cwttau, expy, pnow, pymax, s, ss, sumc, sumcy, sums,
        sumsh, sumsy, swtau, var, wtau, xave, xmax, yy, arg, wtemp;
222     double wi[n+1], wpi[n+1], wpr[n+1], wr[n+1];
223     static double sqrarg;
224
225     double h, yyy;
226
227     *nout = 0.5 * ofac * hifac * n;
228
229     if (*nout > np) {
230         printf("output_arrays_too_short_in_period\n");
231         exit(1);
232     }
233
234     avevar(y, n, &ave, &var);

```

```

235     *sigma2 = var;
236     *aver = ave;
237
238     *pdc = 0;
239     for(j=0; j<n; j++){
240         yyy = y[j] - ave;
241         h += yyy;
242     }
243     *pdc = 0.5 * h * h / var; //Power_of_DC
244
245     xmax = x[n-1];
246     xave = 0.5 * xmax;
247     pymax = 0.0;
248     pnow = 1.0 / (xmax * ofac);
249
250     for (j=0;j<n;j++) {
251         arg = 2* M_PI * ((x[j] - xave) * pnow);
252         wpr[j] = -2.0 * ((sqrarg=(sin(0.5*arg))) == 0.0 ? 0.0 : sqrarg*
                sqrarg);
253         wpi[j] = sin(arg);
254         wr[j] = cos(arg);
255         wi[j] = wpi[j];
256     }
257
258     for (i=0;i<(*nout);i++) {
259         px[i] = pnow;
260         sumsh = sumc = 0.0;
261         for (j=0;j<=n;j++) {
262             c = wr[j];
263             s = wi[j];
264             sumsh += s*c;
265             sumc += (c-s)*(c+s);
266         }
267
268         wtau = 0.5 * atan2f(2.0*sumsh,sumc);
269         swtau = sin(wtau);
270         cwttau = cos(wtau);
271         sums = sumc = sumsy = sumcy = 0.0;
272
273         for (j=0;j<n;j++) {
274             s = wi[j];
275             c = wr[j];
276             ss = s * cwttau - c * swtau;
277             cc = c * cwttau + s * swtau;
278             sums += ss * ss;
279             sumc += cc * cc;
280             yy = y[j]-ave;
281             sumsy += yy * ss;
282             sumcy += yy * cc;
283             wr[j] = ((wtemp=wr[j]) * wpr[j] - wi[j] * wpi[j]) + wr[j];
284             wi[j] = (wi[j] * wpr[j] + wtemp * wpi[j]) + wi[j];
285         }
286

```

```

287         py[i] = 0.5 * (sumcy * sumcy / sumc + sumsy * sumsy / sums) / var;
288
289         if (py[i] >= pymax) pymax = py[*jmax=i];
290
291         pnow += 1.0 / (ofac * xmax);
292     }
293
294     expy = exp(-pymax);
295     *prob = 1.0 - pow(1.0-expy,*nout);
296 }
297
298 void avevar(double data[], unsigned long n, double *ave, double *var){
299     unsigned long j;
300     double s,ep;
301
302     for (*ave=0.0,j=0;j<n;j++){
303         *ave += data[j];
304     }
305     *ave /= n;
306     *var=ep=0.0;
307     for (j=0;j<n;j++) {
308         s=data[j]-(*ave);
309         ep += s;
310         *var += s*s;
311     }
312     *var=(*var-ep*ep/n)/(n-1);
313
314 }
315
316 void significantlevel(double nout, double fap[6]){
317     double recn;
318
319     recn = 1 / nout;
320
321     fap[0] = -log(1.000 - pow(0.999, recn));
322     fap[1] = -log(1.000 - pow(0.995, recn));
323     fap[2] = -log(1.00 - pow(0.99, recn));
324     fap[3] = -log(1.00 - pow(0.95, recn));
325     fap[4] = -log(1.0 - pow(0.9, recn));
326     fap[5] = -log(1.0 - pow(0.5, recn));
327 }
328
329 void count(char *filename, int *cc, int *cr, int *skip){
330     FILE *fp;
331     char buf[2048];
332     int i = 0, k = 0, countc = 0, countr = 0;
333
334     if((fp=fopen(filename, "r")) == NULL){
335         printf("Can not open[count].\n");
336         exit(1);
337     }
338
339     while ((fgets(buf,sizeof(buf),fp)) != NULL){

```

```

340         k++;
341         if(strncmp(buf,"#",1) == 0)continue;
342         *skip = k - 1;
343         break;
344     }
345
346     while(i<2048){
347         if(buf[i] == '\0')break;
348         if(buf[i] == '_')countc++;
349         i++;
350     }
351
352     while ((fgets(buf,sizeof(buf),fp)) != NULL){
353         if(strncmp(buf,"#",1) == 0)continue;
354         countr++;
355     }
356
357     *cc = countc + 1;
358     *cr = countr + 1;
359
360     fclose(fp);
361 }
362
363 void fit(double tofitx[], double tofity[], int cr, double *m, double *l){
364     double a, b, c, d, e;
365     int i;
366
367     a = b = c = d = e = 0;
368
369     for(i=0; i<cr; i++){
370         a += tofitx[i]*tofitx[i];
371         b += tofitx[i];
372         c += 1;
373         d += tofitx[i]*tofity[i];
374         e += tofity[i];
375     }
376     //y = m * x + l
377     *m = (c*d - b*e)/(a*c - b*b);
378     *l = (a*e - b*d)/(a*c - b*b);
379
380 }
381
382 void gnufile(double fap[6], double pxmax, double pymax, int seibun, char *
    filename, char *result, char *gpname, double tognuv[])
383 {
384     FILE *fp;
385     int l = 0, i, j;
386     char s[64];
387     double realmax;
388
389     strcpy(s, filename);
390     strcat(filename, "_lsperiodogram_gnu.txt");
391     if((fp=fopen(filename, "w")) == NULL){

```

```

392         printf("Can not open [gnu].\n");
393         exit(1);
394     }
395
396     strcpy(gpname, filename);
397
398     realmax = pxmax;
399     if(pxmax>0.1) pxmax=0.1; //if large px[]
400
401     fprintf(fp, "reset\nset_term\eps_enhanced\color\
        \"Times-Roman\
        \"_18\nunset_colorbox\nset_keyfont\
        \'Times-Roman,_13\''\nset_yl\
        Normalized Power\"\nset_xl\
        \"frequency [cycle/day]\"\nset_y2l\
        \"false
        -alarm probability\"\nset_y2tics\
        (\"0.001\"_);
402     fprintf(fp, \"%lf\", fap[0]);
403     fprintf(fp, \"_\"_0.005\"_);
404     fprintf(fp, \"%lf\", fap[1]);
405     fprintf(fp, \"_\"_0.01\"_);
406     fprintf(fp, \"%lf\", fap[2]);
407     fprintf(fp, \"_\"_0.05\"_);
408     fprintf(fp, \"%lf\", fap[3]);
409     fprintf(fp, \"_\"_0.1\"_);
410     fprintf(fp, \"%lf\", fap[4]);
411     fprintf(fp, \"_\"_0.5\"_);
412     fprintf(fp, \"%lf\", fap[5]);
413     fprintf(fp, \"_font\
        \'Times-Roman,_14\''\nset_grid_noytics_y2tics_lw_3\n\
        nset_xrange_[0:");
414     fprintf(fp, \"%lf\", pxmax);
415     fprintf(fp, "]"_n#set_xrange_[0:%lf]\nset_yrange_[0:", realmax);
416     fprintf(fp, \"%lf\", pymax);
417     fprintf(fp, "]"_nset_grid_noxtics_x2tics\nset_title\
        \"%s Normalized Lomb-
        Scargle periodogram\"\nset_out\
        \"%s_normalized_periodogram.eps\"_
        nplot\"_\"_\"_\"_);
418
419     if(seibun==1){
420         fprintf(fp, \"%s\"_u_1:3_ti\
            \"%4.2lf km/s\"_w_l\n\", result,
            tognuv[0]);
421     }
422     else{
423         fprintf(fp, \"%s\"_u_1:3_ti \
            \"%4.2lf km/s\"_w_l,\\n\",_result,_
            tognuv[0]);
424         j=1;
425         for(i=1; i<seibun-1; i++){
426             fprintf(fp, \"_\"_u_1:%d_ti \
                \"%4.2lf km/s\"_w_l,\\n\",_i
                +3, tognuv[j]);
427             j++;
428         }
429         fprintf(fp, \"_\"_u_1:%d_ti \
            \"%4.2lf km/s\"_w_l\n\",_seibun+2,_
            tognuv[j]);
430     }
431
432     fclose(fp);
433 }
434

```

```

435 void countname(int *cn){
436     FILE *fp;
437     char buf[64];
438     int countname=0;
439
440     if((fp=fopen("filenamelist.txt","r"))!=NULL){
441         printf("Can not open[namelist].\n");
442         exit(1);
443     }
444
445     while((fgets(buf,sizeof(buf),fp))!=NULL){
446         countname++;
447     }
448
449     *cn=countname;
450
451     fclose(fp);
452 }

```

D.2 プログラムの実行方法

上記のプログラム `lsperiodogram` と上記の書式に従って記録された観測データがあれば、正規化 Lomb-Scargle ピリオドグラムを求めることができる。その際、実行方法は次の4つがある。なお、プログラム `lsperiodogram` を、「`lsperiodogram`」の名前でコンパイルしたとする。

1. 複数の天体を一度に処理する方法。この方法では、正規化 Lomb-Scargle ピリオドグラムを求めたい天体の観測データ名をリスト化したテキストファイル「`filenamelist.txt`」を用意する。例えば、
`4.434+03.rcparam.dat`
`6.18903.rcparam.dat`
`...`
 とリスト化された、「`filenamelist.txt`」が存在すれば、一度に複数の天体を処理可能である。
 なお、`ofac`, `hifac` は1で実行されるため、オーバーサンプリングは行われない。
2. 一つの天体のみを処理する方法。この方法では、プログラムの実行時に、引数として天体のファイル名を記載する。例えば、
`$ ./lsperiodogram 4.434+01.rcparam.dat`
 のように実行する。(1.)と同じく、`ofac`, `hifac` は1で実行されるため、オーバーサンプリングは行われない。
3. (1.)の方法を用いる際、`ofac`, `hifac` を変更する方法。プログラムの実行時に、引数として §3.3 で与えた `ofac`, `hifac` を続けて入力する。例えば、`ofac = 8`, `hifac = 2` とする場合、
`$ ./lsperiodogram 8 2`
 のように実行する。このとき、`nout` 個の結果が格納される (関数 `period` 内の) 配列 `py[ ]` のサイズ `np=SIZE2` が必要な分確保されているか注意する。
4. (2.)の方法を用いる際、`ofac`, `hifac` を変更する方法。プログラムの実行時に、引数として天体の観測データ名、`ofac`, `hifac` を続けて入力する。例えば、`ofac = 8`, `hifac = 2` とする場合、
`$ ./lsperiodogram 4.434+01.rcparam.dat 8 2`
 のように実行する。(3.) 同様に、`np` のサイズ (`SIZE2`) に注意する。

D.3 関数の説明

プログラム `lsperiodogram` に含まれる関数は、正規化 Lomb-Scargle ピリオドグラムを計算する `period`、トレンド成分を最小二乗法により求める `fit` の他に次の関数がある。

関数 `avevar` は、式 (8), (9) で与えられる分散と平均を計算するための関数である。プログラム `lsperiodogram` 中では、作図用に天体の各視線速度成分の平均を求めるとき、関数 `period` 内で Lomb-Scargle ピリオドグラムを正規化する際に用いる分散を求めるとき、§3.2.2 で説明した平均を 0 にするときに用いる平均を求めるときに使う関数である。

関数 `count` は、観測データの行数と列数を数えるための関数である。上記の「(天体名)_rcparam.dat」の書式に従ったデータのみに対応させている。

関数 `significantlevel` は、式 (12) で与えられる閾値 z_0 を、各誤警報確率で求めるための関数である。ここでは、誤警報確率が、0.001, 0.005, 0.01, 0.05, 0.1, 0.5 とした各閾値を導出するようにしている。

関数 `gnupfile` は、gnuplot を用いて作図する際に読み込むファイルを作成するための関数である。ここで作成される gnuplot の load ファイルにより、§4.2 節で示すような正規化 Lomb-Scargle ピリオドグラムのスペクトルを簡単に作図できる。

関数 `countname` は、実行方法 1, または 3. を用いる際に必要なファイル名リスト「`filenamelist.txt`」を読み込み、ファイルの数を数える関数である。ファイルの数を数えることにより、正規化 Lomb-Scargle ピリオドグラムを計算するループ回数を指定する。

関数 `main`^{*13} は、各関数の参照を行ったり、解析ファイルや計算結果を配列に格納したり、テキストファイルに出力する処理を行っている。

D.4 プログラムの内容説明

私の修士論文で作成した C 言語周期解析プログラム Lomb-Scargle ピリオドグラムの内容、使い方等について説明する。行番号はプログラムの行番号に対応させてある。なお、このプログラムは、私のコンピュータ

- Macbook Pro (Retina, 13-inch, Late 2013)
- プロセッサ (CPU) 2.4 GHz Intel Core i5
- メモリ 4 GB 1600 MHz DDR3

での動作を保証している。

【概要】

当プログラムは、バックアップ HDD に「`lsperiodogram.c`」で保存されています。実行方法は 4 つあり、その説明は後述である。プログラムが実行されると、1 天体につき、LS 法の計算結果と

^{*13} 関数 `main` は、C 言語に必ず記述しなければいけない関数である。

gnuplot 作図用のテキストファイルが、それぞれ作成される。また、gnuplot 作図ファイルを簡単に読み込める gnuplot の load ファイルも作成される。

それでは、プログラムの記述を上から順に説明していく。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <math.h>
5 #define SIZE 1028
6 #define SIZE2 SIZE*4 //nout
```

ディレクティブ^{*14}の部分は、`#include`と`#define`がある。`#define SIZE 1028`の数値“1028”は、`rcparam.dat` ファイルを格納するための配列のサイズを決めるものである。そのため、読み込みたい `rcparam.dat` ファイルの行数以上である必要がある。コンピュータのメモリを消費するようであれば小さい数値に変更しても構わないが、`rcparam.dat` ファイルの行数以下であるとプログラムは実行されても結果がデタラメになる。なお、`rcparam.dat` の行数をカウントするプログラム (`countrcparam.c`) は別途作成しているため、配列の設定の目安にしてください。もう一つの `#define SIZE2 SIZE*4` は、ピリオドグラムの計算結果を格納するための配列サイズを決める。ここでは、`SIZE(=1028)` の4倍と設定しているが、これは、修論で説明してあるオーバーサンプリングを行う場合、読み込んだ `rcparam.dat` の行数の整数倍の結果が出力されることへの対応である。コンピュータのメモリを消費する場合や、オーバーサンプリング係数が小さい場合は、数値を小さくして構わない。

```
8 void avevar(double data[], unsigned long n, double *ave, double *var);
9 void period(double x[], double y[], int n, double ofac, double hifac, double px
    [], double py[], int np, int *nout, int *jmax, double *prob, double *sigma2,
    double *pdc, double *aver);
10 void count(char *filename, int *cc, int *cr, int *skip);
11 void fit(double tofitx[], double tofity[], int cr, double *m, double *l);
12 void significantlevel(double nout, double fap[6]);
13 void gnufile(double fap[6], double pxmax, double pymax, int seibun, char *
    filename, char *result, char *gpname, double tognuv[]);
14 void countname(int *cn);
```

関数のプロトタイプ宣言の部分である。関数については、修論にて記載、及び後述してある。

```
16 int main(int argc, char *argv[]){
17     FILE *fp, *fout2, *fname, *fout3, *fout4;
18     int i=0, j=0, k, p=0, seibun, np = SIZE2, nout, jmax, cc, cr, skip,
        length, namei, cn, q;
19     double m, l, avelo, var_dummy, ave, var, prob, effm, pxmax, pymax, ofac,
        hifac, mjdmax, cycle;
20     char filename[64], result[64], name[64], gpname[64], loadfile[600][64],
        buf[512];
```

^{*14} 冒頭の「#」が付く記述の部分

```

21     double rcp[SIZE][64], tofitmjd[SIZE], tofitx[SIZE], tofity[SIZE], toave[
        SIZE], fluxfit[SIZE], toperiod[SIZE], px[SIZE2], py[SIZE2], fap[6],
        kekka[SIZE2][32], pjmax[30], pro[30], pperiod[30], tognuv[30], pron1
        [30];
22     int jmaximum[30], pickup[30], judge[30];
23
24     double variance[30], average[30];
25     double sigma2, pdc, aver;

```

ここから、main 関数が始まる。ここは変数の宣言の部分である。このプログラムはコマンドライン引数^{*15}を持つため、main 関数に int main(int argc, char *argv[]) と引数を書き加えている。変数については、無駄な変数も含まれている可能性があるため、随時変更してください。また、各配列のサイズは余裕を持たせた設定であるため、こちらも明らかに無駄なものは随時変更してください。

```

27     //////////////////////////////////arg////////////////////////////////
28     if (argc != 3 && argc != 4){
29         ofac = 1;
30         hifac = 1;
31         printf("NOW:ofac=1.0_hifac=1.0[default]\n");
32     }
33
34     if (argc == 1){
35         printf("Analyzed_\\"filenamelist.txt\\"\\n");
36     }
37
38     if (argc == 2){
39         sscanf(argv[1], "%s", filename);
40         strcpy(filename, argv[1]);
41         goto point;
42     }
43
44     if (argc == 3){
45         sscanf(argv[1], "%lf", &ofac);
46         sscanf(argv[2], "%lf", &hifac);
47         printf("NOW:ofac=%3.2lf_hifac=%3.2lf\\n", ofac, hifac);
48     }
49
50     if (argc == 4){
51         fp = fopen(argv[1], "r");
52         sscanf(argv[2], "%lf", &ofac);
53         sscanf(argv[3], "%lf", &hifac);
54         printf("NOW:ofac=%3.2lf_hifac=%3.2lf\\n", ofac, hifac);
55         strcpy(filename, argv[1]);
56         goto point;
57     }

```

「arg」で括られたこの部分は、プログラムの実行方法を記述したものである。この部分がコマンドライン引数の設定である。それでは、どのような実行方法を設定してあるかについて、順に説明し

*15 ターミナルでプログラムを実行する際に、引数を渡せること。

ていく。まず,

```
28 if (argc != 3 && argc != 4){
29     ofac = 1;
30     hifac = 1;
31     printf("NOW:ofac=1.0_hifac=1.0[default]\n");
32 }
```

の部分は、実行時の引数が3つ、または4つでない場合に実行する記述である。この場合、オーバーサンプリングが行われないように指定してある。すなわち、ofac, hifac は1で実行される。この指定を設定する理由は、後述のオーバーサンプリングをする実行方法と区別するためである。注意したいのは、C 言語では、当プログラムを例えば「a.out」の名前でコンパイルして実行する際に、

```
$ ./a.out
```

と記述したら、引数は1つであると数えられることである。この場合、「\$./a.out」の後にスペースを入れて何かを入力すると、引数は2つと数えられる。次に、

```
34 if (argc == 1){
35     printf("Analyzed_\n"filenamelist.txt"\n");
36 }
```

の部分は、複数の天体を一度に処理する方法を記述したものである。コマンドライン引数は入力する必要はない。この方法では、Lomb-Scargle 法を実行したい天体の rcparam.dat ファイル名をリスト化したテキストファイル「filenamelist.txt」*16 を用意しておく必要がある。例えば、

```
4.434+03_rcparam.dat
6.189-03_rcparam.dat
...
```

とリスト化された、「filenamelist.txt」が存在すれば、一度に複数の天体を処理可能である。「filenamelist.txt」の簡単な作り方として、ターミナルを用いてリスト化したい rcparam.dat があるディレクトリで、

```
ls *rcparam.dat > filenamelist.txt
```

と実行すれば、作成される*17。なお、この実行方法では ofac, hifac は1となるため、オーバーサンプリングは行われない。また、この実行方法が完了すると、ターミナル上に「Analyzed "filenamelist.txt"」と表示される。

```
38 if (argc == 2){
39     sscanf(argv[1], "%s", filename);
40     strcpy(filename, argv[1]);
41     goto point;
42 }
```

の部分は、一つの rcparam.dat ファイルのみを指定する方法を記述したものである。この方法では、プログラムの実行時に、引数として rcparam.dat ファイル名をスペースを空けて続けて記載する。例えば、

```
$ ./lspriodogram 4.434+01_rcparam.dat
```

のように実行する。この場合も、ofac, hifac は1で実行されるため、オーバーサンプリングは行われない。

*16 この名称指定である

*17 他のやり方を用いた際は、「filenamelist.txt」内の改行コードに注意していただきたい。

```

44 if (argc == 3){
45     sscanf(argv[1], "%lf", &ofac);
46     sscanf(argv[2], "%lf", &hifac);
47     printf("NOW:ofac=%3.2lf_hifac=%3.2lf\n", ofac, hifac);
48 }

```

の部分は、複数の天体を一度に処理する「filenamelist.txt」を読み込ませる方法を用いる際、ofac, hifac を変更する方法の記述である。プログラムの実行時に、引数として修論の §3.3 で与えた ofac, hifac を続けて入力する。例えば、ofac = 8, hifac = 2 とする場合、

```
$ ./lsperiodogram 8 2
```

のように実行する。このとき、nout 個の結果が格納されることになる (period 関数内で用いるの) 配列 py[] のサイズ SIZE2 が np 以上確保されているか注意する^{*18}。Numerical Recipes in C によると、推奨されるオーバーサンプリング係数は 4 以上である。一方、hifac 係数はほとんどの場合で、1 で構わない。^{*19}

```

50 if (argc == 4){
51     fp = fopen(argv[1], "r");
52     sscanf(argv[2], "%lf", &ofac);
53     sscanf(argv[3], "%lf", &hifac);
54     printf("NOW:ofac=%3.2lf_hifac=%3.2lf\n", ofac, hifac);
55     strcpy(filename, argv[1]);
56     goto point;
57 }

```

の部分は、一つの天体のみを指定する方法を用いる際、ofac, hifac を変更する方法の記述である。プログラムの実行時に、引数として天体の観測データ名、ofac, hifac を続けて入力する。例えば、ofac = 8, hifac = 2 とする場合、

```
$ ./lsperiodogram 4.434+01.rcparam.dat 8 2
```

のように実行する。なお、一つの rcparam.dat ファイルを指定する方法を用いると、goto 文により、指定されたポイント「point」まで処理を飛ばす記述がされている。この方法は、実はあまり推奨されないやり方であるため、もっとスマートなやり方があれば変更していただきたい。

それでは、main 関数の次の部分の説明に進む。

```

58 ///////////////fileopen////////////////////
59     if((fname = fopen("filenamelist.txt", "r")) == NULL){
60         printf("filename=");
61         scanf("%s", filename);
62         strcpy(filename, argv[1]);
63         goto point;
64     }

```

ここの記述は、上記の 4 つの方法のうち、「filenamelist.txt」を開く方法を指定したにもかかわらず、「filenamelist.txt」が存在しない場合への対処である。この場合、ファイル名を手打ちで入力するようになっている。入力完了後は、「point」まで飛ぶようになっている。

^{*18} この説明は分かりづらいので、要するにディレクティブで設定する SIZE2 を大きめに確保していれば問題はない。

^{*19} データ数、観測間隔・期間から平均ナイキスト周波数を計算してみれば、ある程度蓄積されたデータであれば hifac は 1 で良いことがわかる。

```
66 countname(&cn);
67 printf("----OPENED_FILENAME(%d)----\n", cn);
```

この部分では、「filenamelist.txt」の行数を数えることにより、何天体分の解析をすればいいのか決めるための countname 関数を用いる。数えられた天体数は、ターミナル上で「OPENED_FILENAME()」の () 内に入る。ここで表示される天体数は、解析がうまくいった天体数であるとは限らない。ここで用いられる countname 関数は、次のような中身である。

```
435 void countname(int *cn){
436     FILE *fp;
437     char buf[64];
438     int countname = 0;
439
440     if((fp=fopen("filenamelist.txt", "r")) == NULL){
441         printf("Can not open[namelist].\n");
442         exit(1);
443     }
444
445     while ((fgets(buf, sizeof(buf), fp)) != NULL){
446         countname++;
447     }
448
449     *cn = countname;
450
451     fclose(fp);
452 }
```

countname 関数では、「filenamelist.txt」の行数をカウントする。すなわち、解析したい天体数を数える処理を行う。そのため、この関数内で改めて「filenamelist.txt」を開く。うまく開けなかった場合は、ターミナルに「Can not open[namelist].」が表示される。メインの部分は、while 文で記述されており、変数 countname を増加させていくことで行数を数える。行数が代入された変数 countname は main 関数に戻すために、ポインタ cn に代入される処理が行なわれる。

それでは、main 関数に説明を戻す。

```
68 for(namei=0; namei<cn; namei++){
69     fgets(filename, sizeof(filename), fname);
70     strtok(filename, "\n");
71     printf("%s\n", filename);
```

ここから比較的長い記述が for 文の中に収められている。その中身を順に説明していく。まず、この for 文 (ループ) は、先ほどの countname 関数によって与えられた cn 回ループする。すなわち、この各ループで rcparam.dat ファイルを一つずつ処理 (ピリオドグラムの計算) していく。ここで引用した「point」までの記述は、「filenamelist.txt」から解析する rcparam.dat ファイル名を読み込み、ターミナルに出力させるためだけの記述である。すなわち、うまく開けたファイルに関してはターミナル上に名前が表示される。しかし、例えば 400 天体分がリスト化された「filenamelist.txt」の 100 番目の rcparam.dat で記録様式等に不備がありうまく開けなかった場

合、この 100 番目のファイル名がターミナルに表示されてプログラムが強制終了する。したがって、99 番目までは解析が終了しているが、101 番目以降が正常なファイルであっても解析はされない。

```
73 point:
74     fp = fopen(filename, "r");
75     length = strlen(filename);
76     strncpy(result, filename, length-12); //_rcparam.dat
77     result[length-12] = '\0';
78     strcpy(name, result);
79     strcat(result, "_lsperiodogram_result.txt");
80     fout2 = fopen(result, "w");
81     if(fp == NULL || fout2 == NULL){
82         printf("FILE_OPEN_ERROR\n");
83         exit(1);
84     }
```

「point」から次の括りまでのこの部分は、主に、「文字」の処理である (string 系というのか?)。はじめ、fp に解析ファイルを開く。次に、変数 length にこのループ回目で開く rcparam.dat ファイル名の文字数を代入する。しかし、どうやら、各 rcparam.dat ファイル名の文字数は「(半角 8 文字)_rcparam.dat」と決まっているみたいなので、あまり意味はない処理である。次の strncpy では、ファイル名のケツ 12 文字 (i.e., "_rcparam.dat") を取り除いた部分 (i.e., 天体名のみ) を配列 result にコピーする。次の「result[length-12] = '\0';」は、天体名しか代入されていない変数 result のケツに、後の処理がエラーにならないように文字コード"\0"を代入する記述である。次の strcpy では、配列 result を配列 name にコピーし、保存しておく。次の、strcat では、結果出力ファイルの名前用に、天体名だけが格納されている配列 result のケツに「_lsperiodogram_result.txt」を書き足す。このように書き足されたファイル名を持つ txt ファイルを、次の fout2 では作成する。残る if 文は、ファイルオープンと作成がうまくいかなかった場合の対処である。

```
85 //count(column*row*#)
86 count(filename, &cc, &cr, &skip);
```

ここでは、このループ回目で扱う rcparam.dat ファイルの行と列数などを数えるための count 関数を用いる。この括り (と言っても一行) で用いられる count 関数は次のような中身である。

```
329 void count(char *filename, int *cc, int *cr, int *skip){
330     FILE *fp;
331     char buf[2048];
332     int i = 0, k = 0, countc = 0, countr = 0;
333
334     if((fp=fopen(filename, "r")) == NULL){
335         printf("Can't open[count].\n");
336         exit(1);
337     }
338
339     while ((fgets(buf, sizeof(buf), fp)) != NULL){
340         k++;
341         if(strncmp(buf, "#", 1) == 0)continue;
```

```

342         *skip = k - 1;
343         break;
344     }
345
346     while(i<2048){
347         if(buf[i] == '\0')break;
348         if(buf[i] == ' ')countc++;
349         i++;
350     }
351
352     while ((fgets(buf,sizeof(buf),fp)) != NULL){
353         if(strncmp(buf,"#",1) == 0)continue;
354         countr++;
355     }
356
357     *cc = countc + 1;
358     *cr = countr + 1;
359
360     fclose(fp);
361 }

```

まず、count 関数の引数の説明をする。char 型 filename には、解析対象の rcparam.dat ファイル名が格納されてある。int 型 cc は、count column の略であり、列数を代入するためにある。int 型 cr は、count row の略であり、行数を代入するためにある。int 型 skip は、解析対象の rcparam.dat ファイルに含まれる「#」で始まる行数を代入するためにある (基本的に冒頭 2 行である)。

count 関数では、解析対象ファイルは数値としてではなく、文字として扱う。そのため、解析対象ファイルの各行を格納するために char 型配列 buf を用意し、そのサイズを 2048 と置いている。天体の成分数が多めのものがある場合は、一行分の数値・スペースを文字として数えた時に 2048 文字を超えるおそれがあるので注意していただきたい。count 関数の主な処理は、はじめ fp に解析対象ファイルを開き、一つ目の while 文で「#」で始まる行数を数える。二つ目の while 文で列数を数え、最後の while 文で行数を数える。列数は、rcparam.dat ファイルの日付や MJD やフラックス密度などの数値の間にあるスペースの数を数える。数えられた列数、行数は、main 関数に戻すために、ポインタ cc と cr に代入される処理が最後に行なわれる。

それでは、main 関数に説明を戻す。

```

87 ////////////////rcparam -> array////////////////////
88     for(i=0; i<skip; i++){
89         fgets(buf,sizeof(buf),fp);
90     }
91
92     for(i=0; i<cr; i++){
93         for(j=0; j<cc; j++){
94             fscanf(fp, "%lf", &rcp[i][j]);
95         }
96     }
97
98     seibun = (cc-4)/3;
99     j = 0; //initialize

```

「rcparam ->array」で括られたこの部分は、解析対象の rcparam.dat ファイルの各数値を、LS 法本編で扱うための 2 次元配列 rcp[i][j] に格納するための記述である。この部分で用いられる各 for 文のループ回数は、先ほどの count 関数で得られた回数だけ行われる。終わりに記述されている、変数 seibun は、速度成分を代入するためにある。また、「j=0」は、直後の本編で再び変数 j を用いるために初期化している。余談だが、このプログラムでは、行を扱うループのカウンタ変数に i、列を変数 j で取り扱うようにしている。

```

101 //////////////////////////////////////////////////LS periodogram loop////////////////////////////////////
102     for(k=0; k<cc; k=k+3){
103         if(k+4 >= cc)break;
104         for(i=0; i<cr; i++){
105             toave[i] = rcp[i][k+4];//column of Vlsr
106             tofity[i] = rcp[i][k+5];//column of Flux
107             tofitmjd[i] = rcp[i][1];//column of MJD
108             tofitx[i] = tofitmjd[i] - tofitmjd[0];//MJD[0]->zero
109         }
110         mjdmax = tofitx[cr-1];//Obs span
111         //////////////////////////////////Cal. average of Vlsr////////////////////////////////
112         avevar(toave, cr, &avelo, &var_dummy);
113         tognuv[j] = avelo;
114         //////////////////////////////////fit////////////////////////////////
115         fit(tofitx, tofity, cr, &m, &l);
116         //////////////////////////////////->toperiod[]////////////////////////////////
117         for(i=0; i<cr; i++){
118             fluxfit[i] = m * tofitx[i];//fit function
119             toperiod[i] = tofity[i] - fluxfit[i];//trend remove
120         }

```

いよいよ本編の始まりである。「LS periodogram loop」で括られた部分の序盤に当たるこの部分では、主に、先ほど rcparam.dat ファイルの中身を全て格納した配列 rcp[i][j] から必要な数値列の抽出、視線速度の平均の計算、トレンド除去のためのフィット、及びトレンド除去を行っている。

まず、「LS periodogram loop」は、速度成分毎に cc 回ループする。ただし、各速度成分は 3 列塊 (Vlsr(km/s), peak(Jy), S/N) で記録されているため、ループ回数は 3 ずつ増加する。cc 回を超えたら break し、「LS periodogram loop」を脱する。配列 toave は、視線速度の平均を求めるために、解析対象ファイルから速度情報のみを抽出・格納する。配列 tofity は、フラックス密度の情報のみを抽出・格納する。配列 tofitmjd は、MJD 情報のみを抽出・格納する。ここで格納された MJD 情報、例えば「56295.0118」など無駄に大きな数値で扱わないようにするため、開始を「0」にするために、配列 tofitx はある。ループを外れた mjdmax には、観測期間が代入される。

「Cal. average of Vlsr」では、配列 toave に格納された視線速度の平均を求めるために、Numerical Recipes in C にて与えられた avevar 関数を用いる。平均を求める理由は、後の gnuplot ファイル作成時に、図の凡例に記載するためだけである。引数として「var_dummy」(分散のダミー)があるのは、分散を求める必要はないが、avevar 関数に受け渡す必要があるためである。求められた平均は、配列 tognuv に格納される。

「fit」では、トレンド成分を求めるために、一次関数フィットを行う。fit の処理内容は、修論 §3.2.1 で説明している通りであるため省略する。

「->toperiod[]」では、先ほど求めたフィット関数を用いてトレンド除去を行う。トレンド除去された新たなフラックス密度情報は、配列 toperiod に格納される。

```
121 ///////////////LS algorithm -> array for result/////////////////
122         period(tofitx, toperiod, cr, ofac, hifac, px, py, np, &nout, &jmax
                , &prob, &sigma2, &pd, &aver);
123
124         jmaximum[j] = jmax;
125         pjmax[j] = py[jmax];
126         pperiod[j] = 1 / px[jmax];
127         pxmax = px[nout-1];
128         pro[j] = prob;
129         cycle = pperiod[j];
130         variance[j] = sigma2;
131         average[j] = aver;
```

本編の最重要部分である Lomb-Scargle 法そのものを記述した部分である。まずは、period 関数の引数の説明をする。

tofitx と toperiod は、それぞれ、開始を「0」とした MJD 情報とトレンド除去されたフラックス密度情報が格納された配列である。cr は、count 関数で与えられた行数である。ofac と hifac は、プログラム実行時に指定した、オーバーサンプリング係数である。px, py は Lomb-Scargle ピリオドグラム の計算結果が格納されて返ってくる。nout は、その計算結果が格納される配列のサイズ、すなわち結果の行数である。jmax は、ピリオドグラムの最大値 (ピーク) がある行番号である。prob は、その最大値 (ピーク) の誤警報確率である。sigma2 は、フラックス密度の分散が代入される (あまり意味はない)。pd は、power DC の略で直流成分のパワーが代入されるが、基本的に 0 であるためあまり意味はない。aver は、フラックス密度の平均が代入される (こちらもあまり意味はない、参考程度である)。

period 関数の呼び出しの後に記述されている計算結果の各代入は、順に、jmaximum が結果が格納される配列 py の何行目にパワーの最大値があるかを格納、pjmax がパワーの最大値そのものの値、pperiod がパワーの最大値がある周波数の逆数 (=周期)、pxmax が出力された周波数の最大値 (=結果の横軸 (周波数軸) の大きさ)、pro がパワーの最大値の誤警報確率、cycle が後の「judge」のために導出周期を再格納、variance、average は分散、平均を格納するものである。

period 関数の中身は次のようになっている。

```
219 void period(double x[], double y[], int n, double ofac, double hifac, double px
        [], double py[], int np, int *nout, int *jmax, double *prob, double *sigma2,
        double *pd, double *aver){
220     int i, j;
221     double ave, c, cc, cwttau, expy, pnow, pymax, s, ss, sumc, sumcy, sums,
            sumsh, sumsy, swtau, var, wtau, xave, xmax, yy, arg, wtemp;
222     double wi[n+1], wpi[n+1], wpr[n+1], wr[n+1];
223     static double sqrrarg;
224 }
```

```

225     double h, yyy;
226
227     *nout = 0.5 * ofac * hifac * n;
228
229     if (*nout > np) {
230         printf("output_arrays_too_short_in_period\n");
231         exit(1);
232     }
233
234     avevar(y, n, &ave, &var);
235     *sigma2 = var;
236     *aver = ave;
237
238     *pdc = 0;
239     for(j=0; j<n; j++){
240         yyy = y[j] - ave;
241         h += yyy;
242     }
243     *pdc = 0.5 * h * h / var; //Power_of_DC
244
245     xmax = x[n-1];
246     xave = 0.5 * xmax;
247     pymax = 0.0;
248     pnow = 1.0 / (xmax * ofac);
249
250     for (j=0;j<n;j++) {
251         arg = 2* M_PI * ((x[j] - xave) * pnow);
252         wpr[j] = -2.0 * ((sqrarg=(sin(0.5*arg))) == 0.0 ? 0.0 : sqrarg*
                sqrarg);
253         wpi[j] = sin(arg);
254         wr[j] = cos(arg);
255         wi[j] = wpi[j];
256     }
257
258     for (i=0;i<(*nout);i++) {
259         px[i] = pnow;
260         sumsh = sumc = 0.0;
261         for (j=0;j<=n;j++) {
262             c = wr[j];
263             s = wi[j];
264             sumsh += s*c;
265             sumc += (c-s)*(c+s);
266         }
267
268         wtau = 0.5 * atan2f(2.0*sumsh,sumc);
269         swtau = sin(wtau);
270         cwttau = cos(wtau);
271         sums = sumc = sumsy = sumcy = 0.0;
272
273         for (j=0;j<n;j++) {
274             s = wi[j];
275             c = wr[j];
276             ss = s * cwttau - c * swtau;

```

```

277             cc = c * cwttau + s * swtau;
278             sums += ss * ss;
279             sumc += cc * cc;
280             yy = y[j]-ave;
281             sumsy += yy * ss;
282             sumcy += yy * cc;
283             wr[j] = ((wtemp=wr[j]) * wpr[j] - wi[j] * wpi[j]) + wr[j];
284             wi[j] = (wi[j] * wpr[j] + wtemp * wpi[j]) + wi[j];
285         }
286
287         py[i] = 0.5 * (sumcy * sumcy / sumc + sumsy * sumsy / sums) / var;
288
289         if (py[i] >= pymax) pymax = py[(*jmax=i)];
290
291         pnow += 1.0 / (ofac * xmax);
292     }
293
294     expy = exp(-pymax);
295     *prob = 1.0 - pow(1.0-expy,*nout);
296 }

```

Numerical Recipes in C に記載されているものと概ね変わらないが、一部変更されている。しかし、Lomb-Scargle ピリオドグラムのアプローチそのものに変更はない。変更した点は、dvector というものの削除や、誤警報確率を近似式 (修論 P.46 の式) ではなく 295 行目にて正確に求めるようにしたことである。また、LS 法の計算に用いる平均と分散を main 関数に返すために 235-236 行目を追加した。238-243 行目は、直流成分の計算である。

main 関数の説明に戻す。

```

133 //////////////judge/////////////////
134             if(prob<=0.0010)pickup[j]=1;//fap<0.001->"1"
135             else{
136                 pickup[j] = 0;
137             }
138             if(prob<=0.0010 && 3*cycle < mjdmax)judge[j]=1;//include 3cycles
               ->"1"
139             else{
140                 judge[j]=0;
141             }
142             ///////////////////////////

```

「judge」で括られたこの部分は、誤警報確率が 0.001 以下であるもの、それに加え導出周期が観測期間に 3 周期含まれているかを判断し、結果出力の txt ファイルに印をつけるためにある。誤警報確率が 0.0001 以下であれば配列 pickup に「1」が入る。また、導出周期の 3 周期分が観測期間に収まっていれば、配列 judge に「1」が入るようになっていく。いずれも、当てはまらない場合は「0」が入る。この「1」と「0」の印は結果 txt ファイルにおいて、各速度成分の頭に記載される。基準となる誤警報確率や何周期分かを更改したい場合は、if 文の () の中身を変更すれば良い。

```

144 for(p=0; p<nout; p++){
145     kekka[p][0] = px[p];

```

```

146             kekka[p][j+1] = py[p];
147         }
148         kekka[nout][j+1] = pdc;
149
150         j++;
151     }
152     //////////////////////////////////LS periodogram loop END////////////////////////////////

```

本編の終盤に当たるこの部分は、計算結果を配列 kekka に格納する処理を記述してある。また、最後にループカウンタを更新するために j を増加させている。一応、148 行にて、直流成分が記述されている。

```

154 significantlevel(nout, fap);//Cal. false alarm prob.
155
156     pymax = pjmax[0];
157     for (j=0; j<seibun+1; j++) {
158         if (pjmax[j] > pymax) pymax = pjmax[j];
159     }

```

significantlevel 関数を用いるこの部分は、各誤警報確率の導出と、後の gnuplot 作図ファイルの作成のためにパワーの最大値を選び出す処理を記述している。

significantlevel 関数の中身は次のようになっている。

```

316 void significantlevel(double nout, double fap[6]){
317     double recn;
318
319     recn = 1 / nout;
320
321     fap[0] = -log(1.000 - pow(0.999, recn));
322     fap[1] = -log(1.000 - pow(0.995, recn));
323     fap[2] = -log(1.00 - pow(0.99, recn));
324     fap[3] = -log(1.00 - pow(0.95, recn));
325     fap[4] = -log(1.0 - pow(0.9, recn));
326     fap[5] = -log(1.0 - pow(0.5, recn));
327 }

```

各誤警報確率 (fap) は、修論の式 (12)

$$z_0 = -\ln \left[1 - (1 - \alpha)^{1/N} \right]$$

を用いて計算される。 α が誤警報確率、 N が独立周波数の数 (nout), z_0 がパワーの値となる。計算された誤警報確率は、この後の gnuplot 作図ファイル作成にも回される。significantlevel 関数の引数についてだが、誤警報確率は Lomb-Scargle ピリオドグラムによって出力された周波数の数 (nout) のみに依存するため、その一つしか引数は渡されていない (配列 fap[] は結果が返される引数である)。reciprocal of n の略である変数 recn は、ただのワンクッションであり、あえて記述する意味はない。

main 関数に説明を戻す。

```

160 ///////////////result array->text file////////////////////
161     fprintf(fout2, "#ofac=%3.2lf_hifac=%3.2lf\n", ofac, hifac);
162     fprintf(fout2, "y(0.001)=_%lf_", fap[0]);
163     fprintf(fout2, "y(0.005)=_%lf_", fap[1]);
164     fprintf(fout2, "y(0.01)=_%lf_", fap[2]);
165     fprintf(fout2, "y(0.05)=_%lf_", fap[3]);
166     fprintf(fout2, "y(0.1)=_%lf_", fap[4]);
167     fprintf(fout2, "y(0.5)=_%lf\n", fap[5]);
168
169     fprintf(fout2, "#nout=_%d\n", nout); //The number of independent freq.
170
171     for(j=0; j<seibun; j++){
172         fprintf(fout2, "#%d_%d", judge[j], pickup[j]);
173         fprintf(fout2, "jmax%2d=_%3d", j+1, jmaximum[j]);
174         fprintf(fout2, "var%2d=_%7.3lf_", j+1, variance[j]);
175         fprintf(fout2, "ave%2d=_%7.3lf_", j+1, average[j]);
176         fprintf(fout2, "pjmax%2d=_%9.6lf_", j+1, pjmax[j]);
177         fprintf(fout2, "fap%2d=_%100.99lf_", j+1, pro[j]);
178         fprintf(fout2, "period%2d=_%lf\n", j+1, pperiod[j]);
179     }
180
181     fprintf(fout2, "#frequency_period_power1_power2...\n");
182
183     fprintf(fout2, "0_0");
184     for(j=0; j<seibun; j++){
185         fprintf(fout2, "_%lf", kekka[nout][j+1]);
186     }
187     fprintf(fout2, "\n");
188     for(p=0; p<nout; p++){
189         fprintf(fout2, "_%lf", kekka[p][0]);
190         fprintf(fout2, "_%lf", 1/kekka[p][0]);
191         for(j=0; j<seibun; j++){
192             fprintf(fout2, "_%lf", kekka[p][j+1]);
193         }
194         fprintf(fout2, "\n");
195     }

```

「result array->text file」で括られたこの部分は、LS ピリオドグラムの計算結果が格納された配列 kekka を txt ファイルに出力する処理の記述である。結果ファイルに何を書き込みたいか制御する場合は、この部分を変更してください。183 行目-186 行目は、直流成分のための記述である。

この部分によって、結果出力ファイルは、

```

#ofac=8.00 hifac=1.00
#y(0.001)= 12.759459 y(0.005)= 11.148022 y(0.01)= 10.452366 y(0.05)= 8.822471 y(0.1)= 8.102721 y(0.5)= 6.219711
#nout = 348
#0 0 jmax 1= 293 var 1= 0.160 ave 1= 1.417 pjmax 1= 11.069361 fap 1= 0.005423 period 1= 29.633910
#0 0 jmax 2= 294 var 2= 0.267 ave 2= 0.768 pjmax 2= 11.650173 fap 2= 0.003034 period 2= 29.533456
#1 1 jmax 3= 293 var 3= 0.193 ave 3= 1.047 pjmax 3= 15.167014 fap 3= 0.000090 period 3= 29.633910
#0 0 jmax 4= 294 var 4= 0.143 ave 4= 1.533 pjmax 4= 4.858713 fap 4= 0.933541 period 4= 29.533456
#frequency period power1 power2 ...
0 0 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000115 8712.369600 0.001588 0.000477 0.000536 0.001255 0.002514 0.001070 0.003492 0.000010 0.001670 0.002488 0.003029
0.000230 4356.184800 0.028334 0.008284 0.009375 0.022235 0.044455 0.018678 0.061730 0.000177 0.029383 0.044214 0.053410

```

.....

のように記載される。

【注意】当プログラムでは、177 行目誤警報確率の表示桁数が 100 桁になっているため、まず然るべき桁数に変更してください。(e.g., `fap%2d=%10.9lf`)

それでは main 関数の次の部分に進む。

```
197 gnufile(fap, pxmax, pyman, seibun, name, result, gpname, tognuv);
198     strcpy(loadfile[q], gpname);
199     q++;
200
201     fclose(fp);
202     fclose(fout2);
203 }
```

ここではまず、gnuplot 作図用ファイルを作成する関数、gnufile 関数を呼び出す。その後、203 行目最後の「}」で「ここから比較的長い記述が for 文の中に収められている。」と前述した for 文が閉じる。

gnufile 関数の中身は次のようになっている。

```
382 void gnufile(double fap[6], double pxmax, double pyman, int seibun, char *
      filename, char *result, char *gpname, double tognuv[])
383 {
384     FILE *fp;
385     int l = 0, i, j;
386     char s[64];
387     double realmax;
388
389     strcpy(s, filename);
390     strcat(filename, "_lspirogram_gnu.txt");
391     if((fp=fopen(filename, "w")) == NULL){
392         printf("Can not open [gnu].\n");
393         exit(1);
394     }
395
396     strcpy(gpname, filename);
397
398     realmax = pxmax;
399     if(pxmax>0.1) pxmax=0.1; //if large px[]
400
401     fprintf(fp, "reset\nset term postscript eps enhanced color\n"Times-Roman\
      "\nset colorbox\nset key font 'Times-Roman, 13'\nset y1\n"
      "Normalized Power\n\nset x1\nfrequency [cycle/day]\n\nset y2\nfalse
      -alarm probability\n\nset y2tics (\"0.001\");
402     fprintf(fp, "%lf", fap[0]);
403     fprintf(fp, ",\n\"0.005\");
404     fprintf(fp, "%lf", fap[1]);
405     fprintf(fp, ",\n\"0.01\");
406     fprintf(fp, "%lf", fap[2]);
407     fprintf(fp, ",\n\"0.05\");
408     fprintf(fp, "%lf", fap[3]);
409     fprintf(fp, ",\n\"0.1\");
410     fprintf(fp, "%lf", fap[4]);
411     fprintf(fp, ",\n\"0.5\");
```

```

412     fprintf(fp, "%lf", fap[5]);
413     fprintf(fp, "\font\ 'Times-Roman, 14'\nset grid noyticks y2tics lw 3\n\
        nset xrange [0:");
414     fprintf(fp, "%lf", pxmax);
415     fprintf(fp, "]\nset xrange [0:%lf]\nset yrange [0:", realmax);
416     fprintf(fp, "%lf", pymin);
417     fprintf(fp, "]\nset grid noxticks x2tics\nset title \"%s Normalized Lomb-
        Scargle periodogram\"\nset out \"%s_normalized_periodogram.eps\"\n
        nplot \"\", s, s);

418
419     if(seibun==1){
420         fprintf(fp, "%s\ " u 1:3 ti \"%.2lf km/s\" w l\n", result,
            tognuv[0]);
421     }
422     else{
423         fprintf(fp, "%s\ " u 1:3 ti \"%.2lf km/s\" w l, \\n", result,
            tognuv[0]);
424         j=1;
425         for(i=1; i<seibun-1; i++){
426             fprintf(fp, "\"\ " u 1:%d ti \"%.2lf km/s\" w l, \\n", i
                +3, tognuv[j]);
427             j++;
428         }
429         fprintf(fp, "\"\ " u 1:%d ti \"%.2lf km/s\" w l\n", seibun+2,
            tognuv[j]);
430     }
431
432     fclose(fp);
433 }

```

実は、完全体ではないので、色付けや各種ラベル等は随時変更していただきたい。それでは、中身の説明をする。はじめの方 389 行目からの string 系関数では、ファイル名などの文字を取り扱っている。その後、「if(pxmax<0.1) pxmax=0.1;」とあるのは、作図した時に周波数 0.1[cycle/day] 以上を表記すると軸が大きくなって図が見づらくなることへの対処であり、出力された周波数が 0.1[cycle/day] を越える場合は、0.1[cycle/day] までで止めておくようにする。もちろん、真の周波数の最大値がわかるように、真の値は変数 realmax に代入される。この処理の後、fprintf 関数が羅列されているが、これこそが gnuplot 作図用ファイルの中身を記述させるための処理である。419 行目からの if 文では、成分数が 1 つの場合とそうでない場合で記述を分ける必要があったため記述している。

main 関数に説明を戻す。

```

204 //////////////////////////////////////////////////gnuplot_load_file/////////////////////////////////
205     fout3 = fopen("periodogram.gp", "w");
206     fprintf(fout3, "reset\n");
207     for(q=0; q<cn; q++){
208         fprintf(fout3, "load \"%s\"\n", loadfile[q]);
209     }
210     fprintf(fout3, "reset\n");
211     //////////////////////////////////////

```

```

212
213         if(fname != NULL)fclose(fname);
214         fclose(fout3);
215         return 0;
216     }

```

この部分で `main` 関数は終わる。「gnuplot_load_file」で括られたこの部分では、「filenamelist.txt」に記載された解析した天体分の gnuplot 作図用ファイルを一挙に読み込める (gnuplot の load) ファイル「periodogram.gp」を作成する。すなわち、該当のディレクトリで、
\$ gnuplot periodogram.gp

と入力すれば、「filenamelist.txt」にリスト化された天体の Lomb-Scargle ピリオドグラムのスペクトル図が一度に作成される。この処理が完了すると、`return 0` でプログラムは終了する。

付録 E 解析天体リスト

今回解析した天体を表 23 に載せた。ただし、データ数が 1 または 2 個であるものは除いている。天体名は略称であり、例えば、G 0.092+06 は G 0.092+0.663 のことである。データ数は、日立アンテナ・モニター観測の開始日 2012 年 12 月 30 日から 2016 年 1 月 10 日の間に観測された数を表す。成分数とは、日立アンテナで同定された視線速度成分の数を表す。

表 23 解析対象天体

天体名	データ数	成分数
G 0.092+06	72	9
G 0.167-04	176	1
G 0.212-00	69	6
G 0.315-02	86	15
G 0.376+00	69	1
G 0.409-05	86	3
G 0.475-00	69	6
G 0.496+01	69	14
G 0.546-08	85	14
G 0.645-00	72	8
G 0.673-00	80	13
G 0.836+01	80	2
G 1.008-02	80	5
G 1.147-01	80	8
G 1.329+01	80	1
G 1.719-00	80	6
G 2.143+00	72	17
G 2.521-02	72	4
G 2.536+01	72	7
G 2.591-00	72	7
G 2.615+01	72	12
G 2.703+00	71	5
G 3.253+00	71	7
G 3.312-03	71	1
G 3.442-03	71	1
G 3.502-02	71	2
G 3.910+00	71	5
G 4.393+00	71	9
G 4.434+01	76	16
G 4.569-00	71	1
G 4.586+00	71	2
G 4.676+02	71	6
G 4.866-01	71	1
G 5.618-00	72	5
G 5.630-02	71	10
G 5.657+04	71	8
G 5.677-00	72	1
G 5.885-03	89	4
G 5.900-04	90	4
G 6.189-03	90	12
G 6.368-00	73	6

天体名	データ数	成分数
G 6.539-01	91	3
G 6.588-01	73	6
G 6.610-00	91	6
G 6.795-02	91	22
G 6.881+00	73	3
G 7.166+01	69	9
G 7.601-01	69	15
G 7.632-01	88	14
G 8.139+02	88	3
G 8.317-00	69	3
G 8.683-03	69	8
G 8.832-00	69	11
G 8.872-04	87	7
G 9.215-02	69	12
G 9.621+01	683	16
G 9.986-00	82	13
G 10.09+07	58	1
G 10.20-03	96	6
G 10.28-01	57	12
G 10.32-01	90	21
G 10.32-02	90	17
G 10.35-01	39	1
G 10.44-00	72	23
G 10.47+00	90	27
G 10.72-03	90	2
G 10.82-01	73	6
G 10.88+01	88	14
G 10.95+00	69	3
G 10.627-0	57	16
G 10.629-0	90	22
G 11.03+00	69	5
G 11.10-01	88	8
G 11.49-14	85	18
G 11.93-01	36	6
G 11.93-06	86	14
G 11.99-02	69	5
G 11.903-0	69	6
G 11.904-0	85	12
G 12.02-00	498	10
G 12.11-01	69	14
G 12.18-01	84	18

天体名	データ数	成分数
G 12.19-00	69	16
G 12.20-01	85	21
G 12.26-00	69	8
G 12.52+00	69	5
G 12.62-00	88	8
G 12.68-01	84	10
G 12.77+01	57	1
G 12.88+04	87	11
G 12.90-00	71	5
G 12.90-02	109	8
G 13.17+00	110	1
G 13.65-05	110	8
G 13.69-01	71	5
G 13.71-00	89	8
G 14.10+00	71	14
G 14.23-05	470	9
G 14.33-06	52	6
G 14.39-00	70	4
G 14.45-01	70	5
G 14.49+00	70	3
G 14.52+01	71	2
G 14.60+00	70	3
G 14.63-05	89	2
G 14.99-01	70	4
G 15.03-06	73	3
G 15.09+01	70	2
G 15.60-02	87	1
G 15.66-04	87	4
G 16.11-03	70	1
G 16.30-01	67	5
G 16.40-01	67	1
G 16.58-00	67	12
G 16.66-03	67	1
G 16.83+00	67	7
G 16.85+06	67	2
G 16.86-21	67	5
G 16.97-00	67	4
G 17.02-00	67	6
G 17.02-24	67	7
G 17.63+01	70	1
G 17.86+00	67	10

天体名	データ数	成分数
G 18.07+00	67	8
G 18.15+00	67	3
G 18.26-02	67	7
G 18.34+17	86	6
G 18.44+00	35	1
G 18.46-00	67	6
G 18.66+00	67	8
G 18.73-02	84	6
G 18.83-03	69	4
G 18.87+00	69	1
G 18.88-04	86	7
G 18.99-02	69	1
G 19.00-00	69	8
G 19.24+02	69	6
G 19.26+03	86	3
G 19.36-00	86	4
G 19.47+01	86	16
G 19.49+01	69	1
G 19.60-02	69	2
G 19.61-01	69	14
G 19.61+00	69	7
G 19.66+01	69	5
G 19.70-02	69	4
G 19.75-01	88	2
G 19.88-05	87	2
G 20.08-01	69	2
G 20.24+00	87	15
G 21.41-02	69	12
G 21.57-00	69	9
G 21.87+00	70	3
G 22.05+02	70	12
G 22.34-01	70	13
G 22.36+00	88	9
G 22.43-01	88	20
G 23.01-04	70	11
G 23.19-03	87	14
G 23.24-02	69	13
G 23.32-02	69	7
G 23.39+01	69	5
G 23.43-01	69	13
G 23.48+00	87	10

天体名	データ数	成分数
G 23.66-01	69	18
G 23.70-01	69	14
G 23.80+04	88	2
G 23.91+00	70	6
G 23.98-00	70	9
G 24.00-01	70	9
G 24.14+00	85	3
G 24.33+01	87	12
G 24.50-00	95	9
G 24.54+03	96	8
G 24.64-03	96	5
G 24.67-01	79	5
G 24.78+00	79	16
G 24.93+00	98	8
G 25.27-04	95	6
G 25.38-01	98	5
G 25.38+00	79	2
G 25.41+01	91	5
G 25.53+03	72	1
G 25.65+10	475	6
G 25.70+00	72	11
G 25.82-01	91	11
G 26.52-02	91	9
G 26.59-00	72	6
G 26.60-02	72	9
G 26.65+00	72	3
G 27.21+02	73	4
G 27.22+01	92	8
G 27.28+01	73	7
G 27.36-01	73	16
G 27.78-02	90	4
G 27.78+00	90	9
G 27.86-02	90	1
G 28.02-04	91	6
G 28.14+00	73	10
G 28.20-00	69	11
G 28.28-03	69	3
G 28.39+00	86	11
G 28.53+01	87	4
G 28.69+04	85	4
G 28.81+03	69	9

天体名	データ数	成分数
G 28.82+04	69	5
G 28.83-02	69	20
G 28.86+00	86	1
G 29.31-01	70	7
G 29.86-00	71	15
G 29.95-00	71	18
G 30.20-01	87	15
G 30.30-01	88	8
G 30.30+00	88	16
G 30.40-02	87	9
G 30.53+00	71	5
G 30.59-00	71	12
G 30.59-01	71	1
G 30.70-00	87	8
G 30.76-00	90	22
G 30.78+02	67	3
G 30.82-00	67	23
G 30.82+02	37	4
G 30.86+01	67	16
G 30.96-01	67	6
G 30.98-00	67	4
G 31.04+03	67	7
G 31.06+00	83	5
G 31.16+00	67	3
G 31.28+00	67	14
G 31.41+03	67	12
G 31.58+00	67	7
G 32.03+00	85	11
G 32.11+00	87	16
G 32.74-00	69	15
G 32.80+01	69	2
G 32.97+00	69	8
G 33.09-00	87	22
G 33.13-00	69	6
G 33.40+00	104	17
G 33.64-02	468	8
G 33.74-01	87	10
G 33.86+00	87	6
G 33.97-00	69	10
G 34.10+00	69	5
G 34.24+01	87	10

天体名	データ数	成分数
G 34.39+02	105	8
G 34.74-00	69	4
G 34.78-13	69	7
G 34.82+03	69	1
G 35.02+03	98	10
G 35.20-07	98	13
G 35.20-17	98	10
G 35.25-02	79	1
G 35.39+00	72	1
G 35.40+00	72	1
G 35.59+00	72	2
G 35.79-01	91	13
G 36.02-02	73	1
G 36.10+05	92	18
G 36.64-02	73	1
G 36.70+00	92	7
G 36.84-00	90	12
G 36.90-04	71	1
G 36.92+04	91	1
G 37.04-00	67	10
G 37.38-00	67	1
G 37.40+15	510	5
G 37.47-01	85	13
G 37.52-01	83	13
G 37.54+02	86	9
G 37.60+04	67	8
G 37.74-01	69	1
G 37.76-01	69	6
G 37.77-02	69	5
G 38.03-03	69	12
G 38.08-02	37	5
G 38.12-02	69	8
G 38.20-00	69	7
G 38.26-00	88	7
G 38.26-02	89	6
G 38.56+01	69	1
G 38.60-02	91	1
G 38.66+00	69	1
G 38.92-03	69	4
G 39.10+04	69	15
G 39.39-01	69	2

天体名	データ数	成分数
G 39.54-03	69	1
G 40.28-02	88	21
G 40.41+07	70	13
G 40.62-01	88	5
G 40.94-00	70	4
G 41.08-01	70	3
G 41.12-01	88	3
G 41.12-02	88	7
G 41.16-02	88	11
G 41.23-02	70	5
G 41.27+03	70	1
G 41.34-01	70	8
G 41.58+00	70	1
G 41.87-01	70	1
G 42.03+01	70	9
G 42.30-03	70	5
G 42.43-02	86	4
G 42.70-01	79	5
G 43.04-04	78	7
G 43.08-00	78	5
G 43.15+00	72	20
G 43.17-00	36	15
G 43.80-01	72	4
G 43.87-07	90	15
G 44.31+00	73	1
G 44.64-05	73	1
G 45.07+01	76	2
G 45.44+00	88	9
G 45.47+00	93	6
G 45.47+01	85	8
G 45.49+01	111	9
G 45.57-01	67	1
G 45.81-03	67	12
G 46.07+02	67	2
G 46.12+03	67	3
G 48.89-01	67	1
G 48.90-02	67	1
G 48.99-03	79	1
G 49.03-10	85	8
G 49.27+03	69	9
G 49.35+04	69	5

天体名	データ数	成分数
G 49.41+03	89	15
G 49.47-03	37	17
G 49.48-04	14	24
G 49.49-03	88	27
G 49.60-02	88	13
G 49.62-03	37	4
G 50.00+05	73	6
G 50.29+06	73	6
G 50.78+01	69	2
G 52.20+07	90	3
G 52.66-10	91	6
G 52.92+04	71	5
G 53.04+01	73	1
G 53.14+00	69	2
G 53.62+00	89	2
G 58.75+06	69	9
G 59.78+00	85	12
G 59.84+06	69	5
G 60.56-01	69	2
G 69.52-09	513	7
G 70.12+17	70	4
G 71.51-03	70	2
G 73.04+18	89	4
G 75.76+03	89	8
G 78.10+36	89	5
G 78.62+09	70	1
G 78.89+07	70	1
G 79.75+09	89	4
G 80.85+04	89	3
G 81.76+05	85	14
G 81.87+07	85	9
G 84.95-06	78	5
G 85.40-00	80	6
G 90.90+15	78	3
G 94.58-17	98	4
G 97.52+31	78	4
G 98.02+14	78	2
G 106.8+53	92	1
G 108.1+55	95	4
G 108.7-09	85	4
G 109.8+21	85	5

天体名	データ数	成分数
G 110.2+26	69	1
G 111.2-07	98	4
G 111.5+07	98	13
G 121.2+06	98	6
G 123.0-63	98	5
G 133.9+10	78	11
G 136.8+11	78	1
G 168.0+08	70	1
G 173.4+24	86	7
G 173.6+28	70	1
G 173.7+23	70	1
G 174.1-00	510	6
G 183.3-05	85	4
G 188.7+10	69	2
G 188.9+08	94	9
G 189.0+07	69	4
G 189.4-01	36	1
G 189.7+03	88	4
G 192.6-00	129	13
G 196.4-16	367	4
G 203.3+20	78	1
G 206.5-16	78	1
G 209.0-19	36	2
G 212.0-07	78	3
G 213.7-12	233	4
G 232.6+09	85	4
G 351.7-05	91	7
G 352.6-10	72	12
G 353.4-03	72	3
G 354.6+04	72	13
G 358.8-00	72	8
G 358.9-00	72	7
G 358.9+00	72	1
G 358.9+01	58	4
G 359.1+00	69	7
G 359.4-04	88	7
G 359.6-02	69	8
G 359.9-04	69	1
G 359.9+01	69	2
IRS22198	739	7

参考文献

- Araya, E. D., Hofner, P., Goss, W. M., et al. 2010, ApJL, 717, L133
Cragg, D. M., Sobolev, A. M., & Godfrey, P. D. 2005, MNRAS, 360, 533
Goedhart, S., Gaylard, M. J., & van der Walt, D. J. 2004, MNRAS, 355, 553
Goedhart, S., Langa, M. C., Gaylard, M. J., & van der Walt, D. J. 2009, MNRAS, 398, 995
Goedhart, S., Maswanganye, J. P., Gaylard, M. J., & van der Walt, D. J. 2014, MNRAS 437, 1808
Breckenridge, S. M.; Kukolich, S. G. 1995, ApJ, 438, 504B
Walsh, A. J.; Burton, M. G.; Hyland, A. R.; Robinson, G. 1998, MNRAS, 301, 640WM
Szymczak, P. Wolak, A. Bartkiewicz, and H. J. van Langevelde 2011, A&A, 531, L3
Szymczak, P. Wolak, A. Bartkiewicz 2015, MNRAS 448, 2284
J.P. Maswanganye, M.J. Gaylard, S. Goedhart, D.J. van der Walt & R.S. Booth 2015, MNRAS 446, 2730
Fujisawa K. et al., 2014, PASJ, 66, 788
Inayoshi K., Sugiyama K., Hosokawa T., Motogi K., Tanaka K. E. I., 2013, ApJ, 769, L20
Press W. H., Teukosky S. A., Vetterling W. T., Flannery B. P., 1992, Numerical Recipes the Art of Scientific Computing, 3rd edn. Cambridge Univ. Press, Cambridge, p. 1003
van der Walt 2011, AJ 141:152
Parfenov & Sobolev 2014, MNRAS, 444, 620
Goedhart, M. Gaylard, J. van der Walt and M. Elitzur 2008, ASPC, 387, 124
Scargle J. D., 1982, ApJ, 263, 835
Lomb N. R., 1976, Ap&SS, 39, 447
Horne, J.H., Baliunas, S.L. 1986, ApJ 302, 757
Cumming, A., Geoffrey W. Marcy, R. Paul Butler 1999, ApJ, 526, 890
Koen C. 1988, ApJ, 348, 700
Schwarzenberg-Czerny A. 1996, ApJ, 460, 107
Frescura, F.A.M., Engelbrecht, C.A., Frank, B.S. 2007
Goedhart, M. Gaylard, J. van der Walt 2007, AMEP, 242
Groth, E, J. 1975, ApJ, 29, 285
Pandian, J.D, Momjian, E., Xu, Y., Menten, K.M., Goldsmith, P.F., 2011, ApJ, 730:55
福井康雄, 他 (2009) 「星間物質と星形成 シリーズ現代の天文学 第 6 巻」 日本評論社
中井直正, 他 (2009) 「宇宙の観測 II 電波天文学 シリーズ現代の天文学 第 13 巻」 日本評論家
日野幹雄 (1977) 「スペクトル解析」 朝倉書店
桜井捷海, 霜田光一 (1984) 「応用エレクトロニクス」 裳華房
前田肇 (1997) 「信号システム理論の基礎」 コロナ社
貴家仁志 (1997) 「デジタル信号処理」 昭晃堂
金城繁徳, 尾知博 (1997) 「例題で学ぶデジタル信号処理」 コロナ社
飯國洋二 (2004) 「基礎から学ぶ信号処理」 培風館
金谷健一 (2003) 「これなら分かる応用数学-最小二乗法からウェーブレットまで-」 共立出版
東京大学教養学部統計学教室 (1991) 「統計学入門」 東京大学出版会
松原望 (2013) 「統計学」 東京図書
森棟公夫, 他 (2008) 「統計学 (改訂版)」 有斐閣
北川源四郎 (2005) 「時系列解析入門」 岩波書店
尾崎統, 北川源四郎 (1998) 「時系列解析の方法」 朝倉書店
P.J. ブロックウェル, 他, 逸見功, 他, 訳 (2004) 「入門時系列解析と予測」 CAP 出版