

# 温度グリーン関数の情報圧縮に基づく高速量子多体計算法

埼玉大学大学院理工学研究科 品岡寛・近野直也  
 東京大学大学院工学系研究科 野本拓也  
 岡山大学異分野基礎科学研究所 大槻純也  
 東京大学物性研究所 吉見一慶

## 1 はじめに

グリーン関数に基づく摂動論や場の量子論は、第一原理計算や量子多体計算の基盤として広く使われている。その中でも、温度 (松原) グリーン関数を用いる虚時間形式は、数値的な取り扱いが簡便であることで知られている。ざっと数えるだけでも、GW, RPA (Random Phase Approximation), FLEX (FLuctuation EXchange), DMFT (Dynamical Mean-Field Theory), 連続時間量子モンテカルロ法, など様々な理論に基づく第一原理計算・モデル計算に用いられている。本稿に興味を持った読者の多くは、このような計算手法に触れたことがある人だろう。もちろん、その応用は物性物理に留まるものではなく、量子化学, 核物理でも広く使われている。

このように幅広い分野で使われているものの、その効率的な数値的取り扱い方はまだ十分確立されていない。例えば、第一原理計算では、桁違いに違うエネルギースケールを同時に扱う必要がある。具体的には、バンド幅 (数~数十 eV) と物理現象が起きる低温 ( $1\text{ K} \simeq 0.1\text{ meV}$ ) などである。このような場合には、計算に現れる虚 (松原) 周波数の数が増大し、計算時間や必要なメモリ量が対処できないほど増加してしまう。

さらに、2 粒子レベルの計算 (バーテ・サルペータ方程式など) は、複数の虚周波数に依存したバーテックス関数を扱う必要がある。そのため、周波数依存性をフルに取り込んだ計算は、簡単なモデル計算の範囲でも極めて計算コストが大きく、現実的な物質への適用が妨げられている。

本稿では、このような問題を解決する、汎用的な温度グリーン関数のデータ圧縮と高速計算手法を紹介する。これらの数値計算理論体系は、著者や共同研究者によってこの数年開発されてきた。2018 年の記事 [1] では、その基盤である温度グリーン関数のコンパクトな基底 (中間表現=Intermediate Representation=IR) を簡単に紹介した。その後、2020 年に入り、IR 基底を使った高速なダイアグラム計算手法が開発されたことを契機に、現在国内外において、物性物理や量子化学分野での応用が急速に広がっている。

本稿は、この数年の急速な理論進展の結果を整理し、1. IR 基底の基礎, 2. 高速計算の基盤であるスパーズサンプリング, 3. 数値計算ライブラリ `irbasis`[2] を網羅的に概観する。これにより、読者が参入する際の助けになることを目的としている。

本稿では、すでに確立された 1 粒子レベル (1 粒子グリーン関数や自己エネルギー) の計算手法を解説する。一方、2 粒子レベルの計算手法 (バーテ・サルペータ方程式など) に関しては、現在活発に開発が進んでおり、§6 において現況に軽く触れたい。

## 2 温度グリーン関数の中間表現 (IR) 基底

温度グリーン関数の中間表現基底とは、2 点相関関数の虚周波数・時間依存性をコンパクトに展開できる汎用的な基底系である [3]。その定義と性質をまとめておこう。

2 点相関関数の一般的な定義は、

$$\hat{G}^\alpha(i\omega) = \int_0^\beta d\tau e^{i\omega\tau} \langle T_\tau A^\alpha(\tau) B^\alpha(0) \rangle, \quad (1)$$

である。ここで、 $\beta = 1/T$  は逆温度、 $\tau$  は虚時間、 $T_\tau$

は時間順序演算子,  $\alpha$  は相関関数の統計性を表し F (フェルミオン) か B (ボソン) である ( $k_B=1$  にとる).  $A$  と  $B$  がオペレータで,  $i\omega$  が統計性  $\alpha$  に対応する虚周波数である. 虚時間で定義された量と区別するため, ハット $\wedge$ を虚周波数空間で定義された量につけた. 1 粒子グリーン関数の場合,  $A$  が消滅演算子,  $B$  が生成演算子である.

レーマン表示の形で, 2 点相関関数に対応したスペクトル関数を

$$\hat{G}(i\omega) = \int_{-\omega_{\max}}^{\omega_{\max}} d\omega' \underbrace{\frac{\omega'^{\delta_{\alpha,B}}}{i\omega - \omega'}}_{K^\alpha(i\omega, \omega')} \rho(\omega'). \quad (2)$$

と定義することが出来る. ここで, スペクトル関数  $\rho(\omega)$  が  $[-\omega_{\max}, \omega_{\max}]$  でのみ有限の値をとるように,  $\omega_{\max}$  を十分大きく取る. また, スペクトル関数は実周波数グリーン関数を使って以下のように定義した:

$$\rho(\omega) = \begin{cases} -\frac{1}{\pi} \text{Im} \hat{G}(\omega + i0^+) & (\alpha = F, \text{フェルミオン}), \\ -\frac{\omega}{\pi} \text{Im} \hat{G}(\omega + i0^+) & (\alpha = B, \text{ボソン}). \end{cases} \quad (3)$$

積分方程式 (2) は, 実周波数空間で定義されたスペクトル関数と, 虚周波数空間で定義された応答関数を結びつけていて,  $K^\alpha(i\omega, \omega)$  がそのカーネルである. なお, 式 (3) において, ボソンに対するスペクトル関数の定義が, 余分な  $\omega$  を含むことに注意されたい\*1.

IR 基底は,  $\beta$  と  $\omega_{\max}$  を決めた上で, このカーネルを特異値分解 (展開)

$$K^\alpha(i\omega, \omega') = - \sum_{l=0}^{\infty} \hat{U}_l^\alpha(i\omega) S_l^\alpha V_l^\alpha(\omega') \quad (4)$$

することで定義される [3]. 虚時間では,

$$\begin{aligned} K^\alpha(\tau, \omega') &\equiv - \sum_{i\omega} \frac{1}{\beta} K^\alpha(i\omega, \omega') e^{-i\omega\tau} \\ &= \sum_{l=0}^{\infty} U_l^\alpha(\tau) S_l^\alpha V_l^\alpha(\omega') \end{aligned} \quad (5)$$

\*1 これに対応して, 式 (2) で定義されたボソンのカーネル  $K^B(i\omega, \omega)$  は, 分子の  $\omega$  線形項の存在のため,  $\omega = 0$  におけるポールの寄与を含まない. 従って, 以下で定義されるボソンの IR 基底はそのような零エネルギー構造を表現できない. 回避策として, IR 基底を補強する方法が [4] で議論されている.

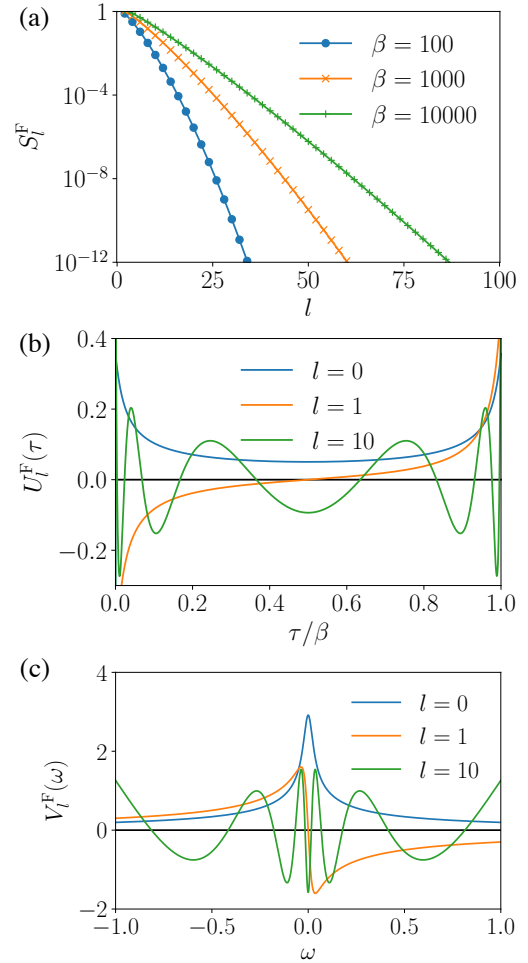


図 1 (a) 様々な  $\beta$  に対する特異値  $S_l^\alpha$ , (b), (c)  $\beta = 100$  における IR 基底関数  $U_l^\alpha(\tau)$  と  $V_l^\alpha(\omega)$ . ここではフェルミオン,  $\omega_{\max} = 1$  の結果を示す. `irbasis`[2] を使って計算した.

と書ける  $[U_l^\alpha(\tau) \equiv \beta^{-1} \sum_{i\omega} \hat{U}_l^\alpha(i\omega) e^{-i\omega\tau}]$ . 我々が IR 基底と呼んでいるのは, 左特異ベクトル  $\hat{U}_l^\alpha(i\omega)$   $[U_l^\alpha(\tau)]$  と右特異ベクトル  $V_l^\alpha(\omega)$  である. IR 基底は, 逆温度  $\beta$ , 統計性  $\alpha$ , エネルギー (周波数) カットオフ  $\omega_{\max}$  のみで決まっていて, 系の詳細に依らないことを強調しておきたい.

図 1 に, フェルミオンに対する特異値  $S_l^\alpha$  と IR 基底関数  $U_l^\alpha(\tau)$ ,  $V_l^\alpha(\omega)$  を示した. IR 基底関数は, 以下にまとめる興味深い性質を持っている\*2.

\*2 性質 2 以外の多くの性質は, カーネルが Total positivity を満たすことで説明できる [5]. なお, IR 基底の解析形はまだ見つかっていない. 数学が得意な方には是非挑戦して欲しい.

**性質 1** 特異値  $S_l^\alpha$  に縮退はなく, 非負で,  $l$  に対して指数関数よりも速く単調減少する.

**性質 2** 数値的に重要な特異値の数 (例えば  $S_l^\alpha/S_0^\alpha \geq 10^{-15}$ ) は, 無次元量  $\Lambda \equiv \beta\omega_{\max}$  によって決まり,  $\Lambda$  に対して高々対数的にしか増加しない.

**性質 3**  $\hat{U}_l^\alpha(i\omega)$ ,  $U_l^\alpha(\tau)$ ,  $V_l^\alpha(\omega)$  は正規直交基底系を張る.

**性質 4**  $U_l^\alpha(\tau)$ ,  $V_l^\alpha(\omega)$  は実関数に選べ, その時  $l$  の偶奇に対して偶・奇関数になる. 対応して,  $\hat{U}_l^\alpha(i\omega)$  は純虚数・実数となる.

**性質 5**  $U_l^\alpha(\tau)$ ,  $V_l^\alpha(\omega)$  は  $l$  個の根を持つ. また,  $\beta\omega_{\max} \rightarrow 0$  の極限 (高温極限) で,  $U_l^\alpha(\tau(x))$ ,  $V_l^\alpha(\omega(x))$  はルジャンドル多項式  $P_l(x)$  に定数倍を除き一致する [ $\tau(x) \equiv \beta(x+1)/2$ ,  $\omega(x) \equiv x\omega_{\max}$  for  $-1 < x < 1$ ] \*3.

さて,  $L$  個の IR 基底関数を使って, 同じ逆温度  $\beta$ , 統計性  $\alpha$  を持つ「任意の」2 点相関関数を展開してみよう:

$$\hat{G}^\alpha(i\omega) = \sum_{l=0}^{L-1} \hat{U}_l^\alpha(i\omega) G_l^\alpha + \epsilon_L. \quad (6)$$

もし, 対応するスペクトル関数が  $[-\omega_{\max}, \omega_{\max}]$  のみで有限であれば, 展開係数  $G_l^\alpha$  は, 「性質 3」を使って,

$$G_l^\alpha = S_l^\alpha \int_{-\omega_{\max}}^{\omega_{\max}} d\omega \rho^\alpha(\omega) V_l^\alpha(\omega) \equiv S_l^\alpha \rho_l \quad (7)$$

と表せる. 式 (7) から,  $G_l^\alpha$  は  $S_l^\alpha$  と少なくとも同程度の速さ, すなわち「性質 1」から指数関数よりも速く減少することが分かる.

では, IR 基底表現のコンパクトさを示す事例を紹介しよう. 図 2 は, 簡単なスペクトル関数  $\rho(\omega)$  の形を仮定した温度グリーン関数を IR 基底で展開して, その展開係数  $\rho_l$ ,  $G_l$  の収束性を示した図である. スペクトル関数の展開係数  $\rho_l$  は, スペクトル関数の形状に強く依存している. 一般的に滑らかな形状の場合には指数関数減衰するが,  $\delta$  関数の場合には減衰しない. 一方, グリーン関数の展開係数  $G_l$  は, スペクトル

ル関数の形状に依らず, 特異値と同程度か, それよりも速く減衰することが分かる. この収束は, カーネルの特異値で決まっていて, システムの詳細 (スペクトル関数の複雑さ) に依らないことを再度強調したい.

式 (7) を見ると,  $G_l$  へ移る際, 高次の  $l$  に対する  $\rho_l$  に特異値  $S_l^\alpha$  がかかることで指数関数的に減衰することが分かる. つまり, スペクトル関数のもつ情報は, グリーン関数に移る際に一部失われ, 逆に温度グリーン関数の数値的データから, 正確なスペクトル関数を復元することは難しい. なぜなら, 逆変換  $\rho_l = (S_l^\alpha)^{-1} G_l$  において,  $G_l$  の誤差が, 小さい特異値で増幅されてしまうからである. それを防ぐ近似的な手法として, 最大エントロピー法, SpM 法 [7, 8] など, 異なる原理に基づく手法が多数提案されているが, ここでは立ち入らない. この点は前回の記事 [1] での解説を参考にしてほしい.

本稿の主眼は, むしろ, 温度グリーン関数を持つ情報が少ないことを積極的に生かし, 虚時間表現で閉じた計算を超高速化することにある. その点で, IR 基底  $\hat{U}_l^\alpha(i\omega)$ ,  $U_l^\alpha(\tau)$  は, その残された情報を効率よく表現できる優れものである.

最後に, IR 基底の低温における優位性をしめす数値例を紹介しよう. 図 3 に, グリーン関数を高精度に表現するのに必要な基底数の  $\beta$  依存性を示した. 具体的には, 図 2(b) と同じスペクトル関数を仮定し,  $G(\tau=0)$  を  $10^{-8}$  の精度で表現するのに必要な基底数を計算した\*4. 図 1(a) から期待されるように, IR 基底の場合, 必要な基底数は高々対数的にしか増えない. 一方, 虚周波数 (Matsubara) 表現, ルジャンドル基底 [6] の場合には,  $O(\beta)$ ,  $O(\sqrt{\beta})$  にそれぞれ比例して増大しており, 極低温にいくほどその差は大きく広がることが見てとれる.

### 3 スパースサンプリング法

#### 3.1 導入: ダイソン方程式

前節で, IR 基底を使えば温度グリーン関数をコンパクトに表現できることが分かった. では, ダイアグ

\*3 虚時間グリーン関数のコンパクトな基底として, DMFT や QMC の文脈でルジャンドル多項式基底 [6] が従来用いられていた. この事実は, ルジャンドル多項式基底が IR 基底の高温極限に対応することを示していて興味深い.

\*4 虚周波数 (Matsubara) 表現の結果では, 非負な虚周波数の数を基底数とした. また,  $G(\tau)$  へのフーリエ変換において, 2 次の高周波数展開を行った.

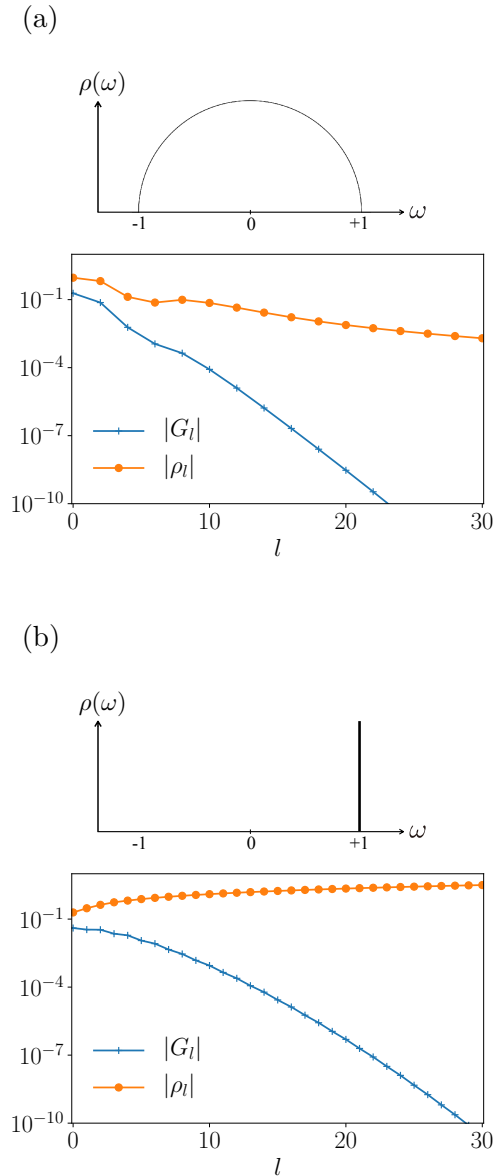


図2 (a) 半円上のスペクトル関数  $\rho(\omega)$ , (b) 絶縁体的なスペクトル関数  $\rho(\omega)$  と対応するグリーン関数の展開係数.  $\beta = 100$ ,  $\omega_{\max} = 1$ ,  $\alpha = \text{F.irbasis}[2]$  を使って計算した. (a) では, 奇数  $l$  に対する  $G_l$ ,  $\rho_l$  は 0 なのでプロットしていない.

ラム展開などで現れるグリーン関数の「演算」を高速に実行できるだろうか? 実は, これは非自明な課題である. 例として, ダイソン方程式を考えてみよう.

$$\hat{G}(i\omega) = \frac{1}{i\omega - \hat{\Sigma}(i\omega)}. \quad (8)$$

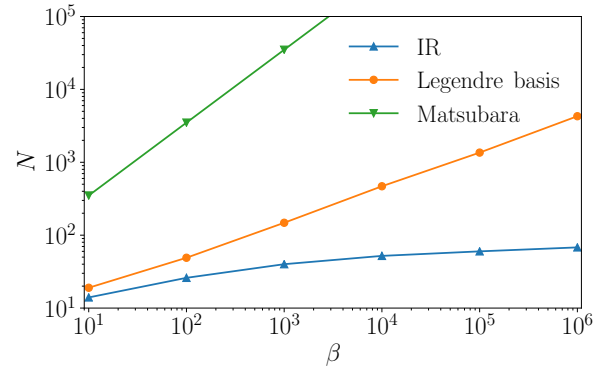


図3 グリーン関数を高精度に表するのに必要な基底の  $\beta$  依存性. 図2(b) と同じスペクトル関数を仮定し,  $G(\tau = 0)$  を  $10^{-8}$  の精度で表現可能な最低基底数を示している.  $\omega_{\max}\beta > 10^4$  において, IR 基底 ( $\omega_{\max} = 1$ ) の優位性が顕著である.

自己エネルギー  $\hat{\Sigma}(i\omega)$  は, IR 基底でコンパクトに展開可能である:

$$\hat{\Sigma}(i\omega) \simeq \sum_{l=0}^{L-1} \Sigma_l \hat{U}_l^F(i\omega). \quad (9)$$

ここでは, 簡単化のためハートリー項を除いた. ここでの問題設定は,  $\Sigma_l$  が数値的に与えられたとき, グリーン関数の展開係数  $G_l$  を効率よく計算できるか, である.

ダイソン方程式を IR 基底内で直接解くのは最善手ではない. ダイソン方程式を  $\hat{G}(i\omega)$  に対する線型方程式に変形すると,

$$\sum_{i\omega'} A_{i\omega, i\omega'} \hat{G}(i\omega') = 1 \quad (10)$$

となる. ただし, 線型演算子は,  $A_{i\omega, i\omega'} \equiv \delta_{i\omega, i\omega'}(i\omega - \hat{\Sigma}(i\omega))$  と定義され, 虚周波数に対して対角成分のみ残る. 原理的には,  $A_{i\omega, i\omega'}$  や  $\hat{G}(i\omega')$  を IR 基底へ基底変換することで, ダイソン方程式を IR 基底による展開係数  $G_l$  に対する線型方程式に帰着することが可能である. しかし, その場合, 線型演算子はもはや対角ではなく,  $L \times L$  の大きさの一次方程式を解く必要がある. そのため, ダイソン方程式の計算コストは, 多バンド系では  $O(N_{\text{band}}^3 L^3) = O(N_{\text{band}}^3 (\log(\beta\omega_{\max}))^3)$  でスケールする. この計算量は,  $\beta\omega_{\max}$  に対して冪関数よりも遅く増大するため, 漸的には従来の方法よりは速い. しかし, 典型

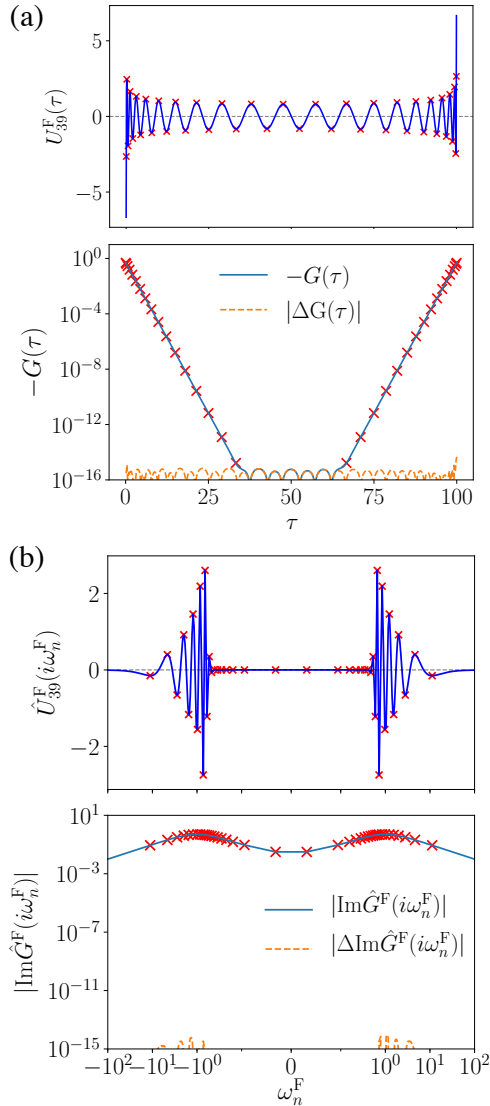


図 4 (a) 虚時間, (b) 虚周波数上のサンプリング点の生成に使う基底関数  $l = 39$  と, サンプリング点上のデータから内挿したグリーン関数 (実線) と厳密な値との差 (破線). 厳密なグリーン関数は, 内挿データと重なって区別できないため示していない. 図 2(a) に示す半円モデルを採用 ( $\beta = 100$ ,  $\omega_{\max} = 1$ ). 文献 [9] の図 2 を改訂したもの.

的なパラメータ,  $N_{\text{band}} = 100$ ,  $L \simeq 100$  において, 従来の手法に対する計算時間の優位性はそれほど大きくはない.

では, IR 基底による表現のコンパクトさと, 虚周波数空間でのダイソン方程式の対角性を同時に享受する方法はあるのだろうか? 次節でその問いに答えよう.

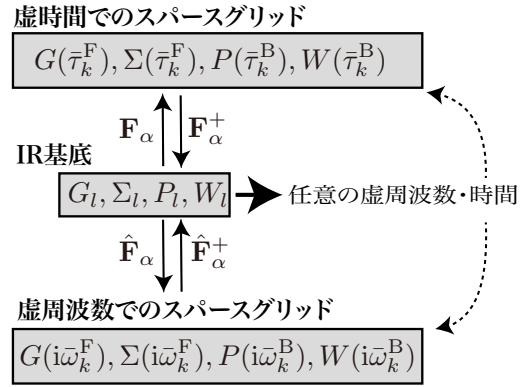


図 5 スパースサンプリング法の概要. IR 基底を経由して, 虚周波数・時間でのスパースグリッドの間でデータを効率的に変換する. IR 基底を経由すれば, 任意の周波数・時間での評価が可能.

### 3.2 スパースサンプリング法とフーリエ変換

スパースサンプリング法 [9] は, 情報のスパース性を巧みに利用して, ダイアグラム方程式の計算量を桁違いに下げることが可能にする. その前に, 虚周波数軸上の情報からどう効率的に情報を引き出すかを議論しよう.

式 (6) の逆変換は,

$$G_l = \sum_{i\omega = -i\infty}^{i\infty} \hat{U}_l^F(i\omega) * \hat{G}(i\omega) \quad (11)$$

である. この式から  $G_l$  を決めるには, 全虚周波数領域における  $\hat{G}(i\omega)$  の情報が必要だが, 実際にはこの式を評価する必要はない. なぜなら, 図 2 で明白なように,  $\hat{G}(i\omega)$  の周波数依存性は, 高々  $L$  個の自由度しか持たないからである. 従って, 適切な  $L$  個の虚周波数  $i\omega$  を選んで, その点での  $\hat{G}(i\omega)$  の値が分かれば,  $L$  個すべての  $G_l$  の値が決定できるはずである. この「適切な周波数」を「サンプリング周波数」と呼ぶことにする.

天降り的であるが, サンプリング周波数として,  $\hat{U}_l^\alpha(i\omega)$  が多項式に類似した符号構造を持つ (「性質 5」) ことを利用し,  $\hat{U}_{L-1}^\alpha(i\omega)$  の極大・極小近傍に選ぶ [9] (図 4):

$$W^\alpha = \{i\omega_1^\alpha, i\omega_2^\alpha, \dots, i\omega_L^\alpha\}. \quad (12)$$

すると, サンプリング周波数上のグリーン関数の値が

	データサイズ	展開の打ち切り誤差	フーリエ変換の計算量
スパースサンプリング法	$L = O(\log(\omega_{\max}\beta))$	$O(e^{-\alpha L})$	$O(L^2)$
従来法 (FFT)	$N_\omega = O(\omega_{\max}\beta)$	$O(1/N_\omega^\gamma)$	$O(N_\omega \log N_\omega)$

 表1 スパースサンプリング法と従来法の比較. 冪  $\gamma (\geq 1)$  は高周波数展開の詳細による.

ら, 最小自乗フィッティングを使って,

$$G_l = \operatorname{argmin}_{G_l} \sum_{i\omega \in \mathcal{W}^\alpha} \left| \hat{G}(i\omega) - \sum_{l=0}^{L-1} \hat{U}_l^\alpha(i\omega) G_l \right|^2$$

$$= (\hat{\mathbf{F}}_\alpha^+ \hat{\mathbf{g}})_l \quad (13)$$

のように  $G_l^\alpha$  を簡単に計算できる. ここで,  $\hat{\mathbf{F}}_\alpha^+$  は, 以下で定義される  $\hat{\mathbf{F}}_\alpha$  のムーア-ペンローズの擬似逆行列である.  $\hat{\mathbf{F}}_\alpha$  はサンプリング周波数での基底関数の値から構成される行列で,  $(\hat{\mathbf{F}}_\alpha)_{kl} = \hat{U}_l^\alpha(i\omega_k^\alpha)$  と定義される. 一方,  $\hat{\mathbf{g}}$  はフィットするグリーン関数の値からなるベクトルで,  $(\hat{\mathbf{g}})_k = \hat{G}^\alpha(i\omega_k^\alpha)$  と定義される.  $\hat{\mathbf{F}}$  は高々  $100 \times 100$  程度の小さな行列であり, 上記のようにサンプリング周波数を選ぶ限り, 経験的に良条件である\*5. 一度  $\hat{\mathbf{F}}^+$  を計算しておけば, 式 (13) は単なる行列-ベクトル積なので, 計算量は  $O(L^2)$  で, BLAS などの高速なライブラリが利用可能である.

虚時間における同様のサンプリング点も, 虚時間での IR 基底  $U_l^\alpha(\tau)$  の符号構造を考えることで構成できる. 結果だけ書いておこう:

$$G_l^\alpha = \operatorname{argmin}_{G_l} \sum_k \left| \hat{G}^\alpha(\bar{\tau}_k^\alpha) - \sum_{l=0}^{L-1} \hat{U}_l^\alpha(\bar{\tau}_k^\alpha) G_l^\alpha \right|^2$$

$$= (\mathbf{F}_\alpha^+ \mathbf{g})_l \quad (14)$$

ここで,  $\bar{\tau}_k^\alpha$  がサンプリング点であり,  $(\mathbf{F}_\alpha)_{kl} = U_l^\alpha(\bar{\tau}_k^\alpha)$  と定義する.

では, スパースサンプリング法によって, 実際に展開係数  $G_l$  を高精度に計算できることを数値的に見てみよう. 図 4(a), (b) に,  $\omega_{\max} = 1$ ,  $\beta = 100$  におけるサンプリング点と, サンプリング点の生成に使用した基底関数を虚時間・虚周波数の場合にそれぞれ

示した. 基底関数が符号変化を持つことが明瞭に見て取れる. 虚時間のサンプリング点は, グリーン関数が構造を持つ  $\tau = 0, \beta$  近傍で密に分布している [図 4(a)]. 一方, 虚周波数では, サンプリング点の分布は低周波数において密で, グリーン関数が構造を持たない高周波数で疎になる [図 4(b)]. また, 図 4(a), (b) には, 図 2(a) と同じ模型に対して, サンプリング点上のグリーン関数の値から IR 基底の展開係数を見積もったあと, 虚時間・虚周波数でデータを「内挿」した結果を示してある. 内挿したグリーン関数は, 厳密な値と 15 桁程度の精度で一致しており, スパースサンプリング法の精度や数値的安定性が分かる.

ここで,  $U_{L-1}^\alpha(\tau)$ ,  $U_{L-1}^\alpha(i\omega)$  の構造から決めたサンプリング点を使って,  $l < L-1$  の展開係数  $G_l$  をなぜ精度良く決められるのかと思う読者もいるだろう. 実は,  $l < L-1$  の IR 基底関数の根 (ゼロ点) の分布は常に,  $l = L-1$  の基底関数よりも疎である (図 1)\*6. したがって,  $l = L-1$  の基底関数から決めたサンプリング点は,  $l < L-1$  の基底関数の符号変化を捉えており, それら低次の基底関数に対する良いサンプリング点にもなっている.

これにより, スパースサンプリング法の全ての部品が揃った. 以降, 虚周波数, 虚時間におけるサンプリング点のことを「スパースグリッド」と呼ぶことにしよう\*7. 図 5 に示すように, 虚周波数, 虚時間におけるスパースグリッド, IR 基底の間で自由にデータを変換できるようになった. 一旦 IR 基底での展開係数を求めれば, 任意の周波数, 時間においてその値を評価することが出来る. その点で, スパースサンプリング

\*5 式 (13) の数値的安定性は,  $\hat{\mathbf{F}}$  の条件数 (最大特異値と最小特異値の比) で決まる. 本稿で説明した方法でサンプリング周波数を選ぶことで, 条件数を小さく抑えることが出来る. 詳しい議論は [9] の Appendix B を参照して欲しい.

\*6 より正確に言うと,  $l' < l$  に対して,  $U_{l'}^\alpha(\tau)$  の隣り合う 2 つの根の間には,  $U_l^\alpha(\tau)$  の根が 1 つ以上含まれる.  $l = l' + 1$  の場合には, 数学的に証明が可能である. 一般の場合も, これまでの数値実験から反例は得られていない.

\*7 同様のスパースグリッドの提案が他のグループから独立にほぼ同時に行われた [10].

グは、物理的な根拠に基づいたグリーン関数の内挿法であるともいえる。これを利用すれば、式 (16) のダイソン方程式は、サンプリング周波数上でのみ解けば十分であることが分かる。なぜならスパースグリッド上の値から、 $G_l$  すなわち正確な虚周波数/時間依存性が再現できるからである。

従来、虚周波数・時間の中でデータを変換したいときは、高速フーリエ変換を使うのが定石であった。表 1 に従来法とスパースサンプリング法の比較をした。データサイズはさることながら、スパースサンプリング法では、計算量は、逆温度  $\beta$  やバンド幅  $\omega_{\max}$  に対して冪関数よりも遅い増大しか示さない点が大きな特徴である。

なお、本稿では、IR 基底に焦点を当てているが、スパースサンプリングを従来の多項式基底と組み合わせることもできる [9]。

## 4 応用計算例の紹介

前章では、適切に選んだ少数のサンプリング点上でのグリーン関数の値から、IR 基底関数による展開係数が高精度に計算できることを示した。これにより、虚時間・虚周波数空間の中で高速にデータを変換することが可能になる。本章では、スパースサンプリング法を使って、どのようにダイアグラム方程式の解法を効率化するのか、実例を通してみていこう。

### 4.1 希ガス原子・シリコン固体: 自己無撞着 GW 計算と 2 次摂動 (GF2)

スパースサンプリング法を提案した論文 [9] から、自己無撞着 GW 計算, 2 次摂動 (GF2) を使ったベンチマーク計算の結果を紹介しよう。この論文では、数値計算の安定性・精度を確かめるため、物理的には「つまらない」が、技術的に難しい希ガス原子、およびシリコン固体を選んだ。今回の記事では、希ガスの結果を紹介するため、シリコン固体の結果はオリジナル論文を参考にして欲しい。

以下のハミルトニアンを考える。

$$H = \sum_{ij\sigma} h_{ij} c_{i\sigma}^\dagger c_{j\sigma} + \frac{1}{2} \sum_{ijkl} \sum_{\sigma\sigma'} V_{ijkl} c_{i\sigma}^\dagger c_{k\sigma'}^\dagger c_{l\sigma'} c_{j\sigma}. \quad (15)$$

ここで、 $h_{ij}$  は非相互作用ハミルトニアン、 $V_{ijkl}$  は

クーロン積分を表し、 $i, j, k, l$  が軌道の足、 $\sigma, \sigma'$  がスピンの足である。この時、ダイソン方程式は

$$\hat{G}(i\omega_n^F) = [(i\omega_n^F + \mu)I - F - \hat{\Sigma}(i\omega_n^F)]^{-1} \quad (16)$$

と与えられる。ここで、フォック行列  $F = h + \Sigma^{\text{HF}}$  はハートリー・フォック項 (自己エネルギーの静的部分)

$$\Sigma_{ij}^{\text{HF}} = (2V_{ijkl} - V_{ilkj})\rho_{kl}, \quad (17)$$

を含む (以降アインシュタインの縮約記法を使う)。自己エネルギーの動的な部分  $\hat{\Sigma}(i\omega_n^F)$  の選択で、異なる近似が可能である。2 次摂動では、

$$\tilde{\Sigma}_{ij}(\tau) \simeq \Sigma_{ij}^{(2)}(\tau) = -G_{kl}(\tau)G_{qm}(\tau)G_{np}(-\tau) \times V_{ikpq}(2V_{ljmn} - V_{mjln}), \quad (18)$$

GW 近似では、

$$\tilde{\Sigma}_{ij}(\tau) \simeq \Sigma_{ij}^{\text{GW}}(\tau) = -G_{lk}(\tau)\tilde{W}_{ilkj}(\tau), \quad (19)$$

と選ぶ。ここで、 $\tilde{W} = W - V$  は遮蔽された相互作用  $W$  の周波数依存部分で、

$$\hat{W}_{ijkl}(i\omega_n^B) = V_{ijkl} + V_{ijpq}\hat{P}_{pqsr}(i\omega_n^B)\hat{W}_{rskl}(i\omega_n^B), \quad (20)$$

と定義される。ここで、裸の分極関数は以下のように与えられる。

$$P_{ijkl}(\tau) = -G_{il}(\tau)G_{jk}(-\tau). \quad (21)$$

式 (16)–(21) は、虚周波数もしくは虚時間において対角である。従って、それぞれの右辺に現れる応答関数の IR 基底による展開係数が分かっているれば、左辺の値をサンプリング点上で素早く計算した後、直ちに IR 基底へ変換できる。このように、IR 基底を経由することで、自己無撞着計算をデータをコンパクトに保ったまま実行できる。

図 6 は、希ガス原子に対する全電子基底 (cc-pVDZ) を使った計算結果を示している ( $\beta = 1000 E_h^{-1}, T \simeq 316 \text{ K}$ )。希ガス原子は、深いコア状態のため、従来の多項式基底では全電子計算が難しく、有効ポテンシャルを導入しても数千もの基底が必要になる。しかし、IR 基底とスパースサンプリングを組みあわせることで、高々 100 個程度の基底で、全

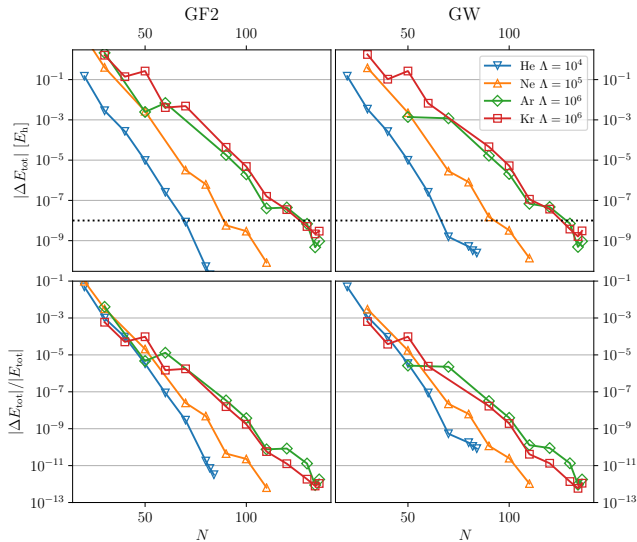


図6 希ガス原子に対する GF2 (左図), GW (右図) 計算における全エネルギーの収束性. 横軸は IR 基底の数. 上の段はエネルギーの絶対誤差 (単位はハートリー  $E_h$ ), 下の段は相対誤差を示している. 文献 [9] より.

電子計算を実行し, しかも基底数に対して収束した結果を得られる.

基底数  $L$  を増やしていくと, 全エネルギーが指数関数的に収束し, 容易に 10 桁程度収束した結果が得られる. 原子番号が増えるほど, コア状態が深くなるが, IR 基底の場合,  $\Lambda = \beta\omega_{\max}$  を増やすことで容易に対応できる. フェルミオン, ボソン両方の応答関数を扱う GW 法の結果も, フェルミオンのみを含む GF2 と同等の収束性を示し, 数値的な安定性の高さが分かる.

#### 4.2 Migdal-Eliashberg 方程式に基づく超伝導転移温度計算

近年, 従来型 (フォノン媒介) 超伝導体の転移温度 ( $T_c$ ) を第一原理計算から予測可能であることは, 読者もお聞き及びと思う. 主な理論手法として, 超伝導密度汎関数理論や Migdal-Eliashberg 理論に基づくアプローチがあるが, 後者は電子-フォノン相互作用による電子の有効質量の繰り込みを自己無撞着に決定できるという利点を持つ. 一方, 現実的な転移温度予測には, フォノン媒介の引力とクーロン斥力という 2 桁も 3 桁も異なるエネルギースケールの物理を同時に取り扱う必要があるため低温で極めて困難にな

る. 本節では IR 基底を用いて, この問題に取り組んだ最近の研究 [11] を紹介する.

Migdal-Eliashberg 理論の枠組みでは, 以下の線形化ギャップ方程式を解くことで  $T_c$  を求める.

$$\tilde{\lambda}\Delta_i(i\omega_n) = -T \sum_j K_{ij}(i\omega_n - i\omega'_n) F_j(i\omega'_n). \quad (22)$$

ここで, 添え字  $i$  は波数とバンドのインデックスを表し,  $\Delta_i$  は超伝導ギャップ関数,  $K_{ij} = K_{ij}^{\text{el-ph}} + K_{ij}^{\text{el-el}}$  は電子格子および (遮蔽された) クーロン相互作用を表す. 異常グリーン関数  $F_i$  は,

$$F_i(i\omega_n) = |G_i(i\omega_n)|^2 \Delta_i(i\omega_n), \quad (23)$$

と表されるため, 式 (22) は行列の固有値問題となっており,  $\tilde{\lambda}$  はその固有値である.  $K_{ij}^{\text{el-ph}}$  による自己エネルギー補正を含んだ  $G_i$  を用いて式 (22) を解き,  $\tilde{\lambda}$  の最大値が 1 となる温度から  $T_c$  を決める, というのが理論のおおまかな枠組みとなっている.

この問題は, ボソンとフェルミオンのグリーン関数の畳み込み積分が虚時間で対角であるから, IR 基底とスパースサンプリング法が適用可能である. 一般に, 現実的な物質は多数のバンドを含んでおり, 低温では収束に必要な波数点の数も膨大であるから, 従来の手法では式 (22) の計算はもとより,  $K_{ij}$  を保持しておくためのメモリ確保すら難しかった. IR 基底を用いれば, 表 1 に示すような, 劇的な改善が期待される.

実際に線形化ギャップ方程式の固有値  $\tilde{\lambda}$  を BCC-Nb および高圧下 (250GPa) での立方晶 LaH<sub>10</sub> に対して計算した例を図 7 に示す. 計算した温度はそれぞれ, 19.7 K および 271.6 K で, それぞれの転移温度より少し高温に設定してある. 図 7(a) は従来法 (虚周波数 +FFT) を用いたものであるが, 高温で計算している LaH<sub>10</sub> は  $N_\omega = 512$  でだいたい収束しているものの, Nb の計算では  $N_\omega = 4096$  でもまだ収束しているか判断できない. 大雑把な見積もりとして, バンドの上限を  $\omega_{\max} \sim 10$  eV とすると,  $\beta\omega_{\max} \simeq 10^4$  であるから, きちんと収束させるにはより多くの  $N_\omega$  が必要だと考えられる. 一方, 図 7(b) に示す IR 基底 + スパースサンプリングの方法だと,  $\Lambda = 10^4$  ですでに完全に収束した値が得られている. 例として



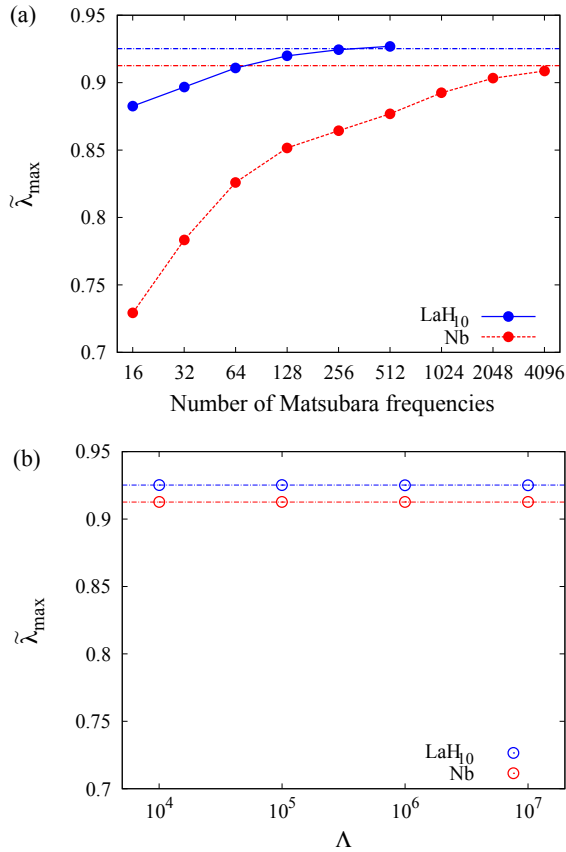


図7 転移温度直上における、線形化されたギャップ方程式の固有値の最大値  $\tilde{\lambda}_{\max}$  の収束性. (a) 虚周波数+FFT, (b) IR 基底 + スパースモデリング. 図中の鎖線は、IR 基底を使って得られた値を示す. 文献 [11] より.

$N_{\omega} = 4096$  と  $\Lambda = 10^5$  の 2 つの計算を比較すると、基底の数は  $L = 138$  で 30 倍程度メモリを節約でき、実際の計算速度も 20 倍程度改善されることが分かった. 今回の計算では電子間相互作用として静的な遮蔽クーロン相互作用を用いているが、乱雑位相近似でプラズモン効果を取り入れる拡張も可能である.

#### 4.3 リヒテンシュタイン公式に基づく磁気相互作用の計算

よく知られているように、密度汎関数理論は基底状態に対する理論であり、そのままでは有限温度の性質を求めることはできない. 磁性体に関して言えば、絶対零度での磁気モーメントの大きさを予測できても、転移温度は計算できない. この問題に対するアプローチにもいくつか種類があるが、電子状態計算の結果から有効スピン模型を導出し、スピン模型に対す

る計算 (平均場近似・モンテカルロ計算など) で転移温度を求める, という手軽で良く用いられる方法である. スピン模型へのマッピングの中でもリヒテンシュタイン法と呼ばれる手法 [12] では, IR 基底 + スパースサンプリング法によって計算が大幅に効率化されることが分かっており, 本節で簡単に紹介する [13, 14, 15].

この理論では, ハイゼンベルグ型のスピン模型  $H = -2 \sum_{\langle i,j \rangle} J_{ij} \mathbf{S}_i \cdot \mathbf{S}_j$  の相互作用  $J_{ij}$  を以下の式で計算する.

$$J_{ij} = -\frac{T}{2} \sum_{\omega_n} \text{Tr}[G_{ij}(i\omega_n)M_j G_{ji}(i\omega_n)M_i]. \quad (24)$$

ここで,  $G_{ij}$  は密度汎関数理論から導出された有効強束縛模型に対するグリーン関数である.  $i, j$  はサイトの添字であり, それぞれのサイトに属する軌道数を  $N_{\text{orb}}^i, N_{\text{orb}}^j$  としたとき,  $G_{ij}$  は  $N_{\text{orb}}^i \times N_{\text{orb}}^j$  行列である.  $M_i$  はスピンの微小回転によるハミルトニアンへの摂動項であるが, ここでは単に  $\omega_n$  に依存しない  $N_{\text{orb}}^i \times N_{\text{orb}}^i$  の定数行列とみなしてほしい. 式 (24) の  $\text{Tr}$  は軌道成分に関するトレースを表す. この項の計算に必要な計算量を考えてみると, グリーン関数の波数空間から実空間へのフーリエ変換が支配的であり, 通常虚周波数メッシュを用いると,  $O(cN_{\omega})$  程度である. ここで係数因子  $c$  は一次元方向の波数メッシュの数を  $N_k$  としたとき,  $c = O((N_k \log N_k)^3 N_{\text{orb}}^2)$  程度であるため, 軌道数が多い複雑な模型となると, それなりの計算コストがかかる. 一方, グリーン関数の計算をスパースグリッド上だけに限れば, この計算量は  $O(cL)$  程度となり, 低温での計算量は圧倒的に少なくなる. 実際の計算では, スパースグリッド上で式 (24) のトレースまで計算してしまい, 最後に IR 基底へ変換する. その後, 虚時間  $\tau = 0$  での値を IR 基底での展開係数から復元してやることで式 (24) が得られるが, この方法では基底の変換にかかるロスほとんど無視でき, 純粋に  $N_{\omega}/L$  倍だけ計算時間が削減できる.

今回はリヒテンシュタイン法を例に取ったが, 一般に二次のハミルトニアン  $H = \sum_{12} A_{12} c_1^{\dagger} c_2$  で記述される系の自由エネルギー  $F[A]$  について,  $A$  に関する微分  $\delta_{A_{12}} F, \delta_{A_{12}} \delta_{A_{34}} F, \dots$  を求めたい場合に, こ

の手法を適用することが出来る。この計算は、もともと前節で紹介したような、いわゆる“重い計算”ではないが、IR 基底が日常的な計算ツールとして利用できることを示す良い例である。今まで見てきたように IR 基底を用いた計算の収束性は極めて良いため、収束性のチェックに必要なパラメータを実質 1 つ減らすことができ、これは数値計算を用いる研究においては、いつでも非常に大きなメリットとなる。

#### 4.4 その他応用

本稿で紹介したもの以外にも、FLEX 近似による超伝導転移温度計算 [16], self-energy embedding 法に基づく NiO 化合物などの第一原理計算 [17], NdNiO<sub>2</sub> の磁気相互作用定数の第一原理的見積もり [15], など第一原理計算分野での応用が進んでいる。また、基底数に対する計算精度の収束が指数関数的であることは、通常のモデル計算においても有用で、自己エネルギーのデノイジング [18], 量子不純物問題の離散化 [19] などに使われている。

## 5 irbasis ライブラリ

IR 基底やスパースサンプリング法によるダイアグラム計算の基礎をこれまでの章で紹介してきた。基底の生成やサンプリング点の選択などの詳細を知らなくても、これらの新しい技術が簡単に使えるように、すでにライブラリ `irbasis` [2] が整備されている。ここでは `irbasis` の概要と、簡単な使い方を説明しよう。`irbasis` は、Python および C++ から利用できるが、今回はスパースサンプリングが実装されている Python 版を紹介する。基底関数の値、サンプリング点の情報をファイルに一旦書き出せば、スパースサンプリングを使ったダイアグラム計算は任意のプログラミング言語で可能である (上で紹介した超伝導計算は Fortran で行った)。

### 5.1 インストール

`irbasis` は PyPI に登録されているので、インストールは極めて簡単で、シェルから以下のコマンドを実行すれば良い。

```
$ pip install -U irbasis
```

U オプションは、`irbasis` がインストール済みの場

合、最新版に更新するオプションである。

### 5.2 無次元表現

IR 基底は、ある種の積分方程式を解くことで計算できるが [20], 小さな特異値に対する基底関数を精度良く計算するには、多倍精度演算が必要で、実用計算の度に積分方程式を解くのは今のところ現実的ではない。そこで、`irbasis` は、典型的なパラメータ  $\Lambda \equiv \beta\omega_{\max} = 10, 10^2, \dots, 10^7$  と  $s_l^\alpha/s_0^\alpha > 10^{-15}$  に対して、事前に計算された基底関数の高精度データが収録されている。`irbasis` は無次元表示  $u_l^\alpha(x)$ ,  $v_l^\alpha(y)$ ,  $s_l^\alpha$  を使っていて [2], 元の表示と

$$U_l^\alpha(\tau) = \sqrt{\frac{2}{\beta}} u_l^\alpha(x(\tau)), \quad (25)$$

$$V_l^\alpha(\omega) = \sqrt{\frac{1}{\omega_{\max}}} v_l^\alpha(y(\omega)), \quad (26)$$

$$S_l^\alpha = \sqrt{\frac{\beta\omega_{\max}^{1+2\delta_{\alpha,B}}}{2}} s_l^\alpha, \quad (27)$$

$$\hat{U}_l^\alpha(i\omega_n) = \sqrt{\beta} u_{nl}^\alpha, \quad (28)$$

の関係にある。なお、 $x(\tau) = 2\tau/\beta - 1 \in [-1, 1]$ ,  $y(\omega) = \omega/\omega_{\max} \in [-1, 1]$  で、

$$u_{nl}^\alpha \equiv \frac{1}{\sqrt{2}} \int_{-1}^1 dx e^{i\pi\{n+(1/2)\delta_{\alpha,F}\}(x+1)} u_l^\alpha(x) \quad (29)$$

と定義した。なお、IR 基底のゲージ自由度は、 $u_l^\alpha(1) > 0$  に選び、 $u_l^\alpha(x) = (-1)^l u_l^\alpha(-x)$ ,  $v_l^\alpha(y) = (-1)^l v_l^\alpha(-y)$  が成り立つ。また、 $u_{nl}^\alpha$  を高精度に計算する関数が実装されているので、数値積分をユーザが実行する必要はない。

### 5.3 基底関数の計算

以下に、 $\Lambda = 1000$  (フェルミオン) のデータをロードして、いくつかの  $x$ ,  $y$ , 虚周波数  $n$  における基底関数を計算するコード例 [21] を示した。`load` は基底関数データをメモリにロードし、基底オブジェクトを作成する。その基底オブジェクトのメンバ関数を呼び出すことで、様々な値が評価できる。例えば、`sl` は  $s_l^\alpha$  を一次元配列として返す。また、`ulx`, `vly` はそれぞれ  $u_l^\alpha(x)$  と  $v_l^\alpha(y)$ , `d_ulx`, `d_vly` はそれらの (高次) 導関数  $\frac{\partial^k u_l^\alpha(x)}{\partial x^k}$ ,  $\frac{\partial^k v_l^\alpha(y)}{\partial y^k}$ , `compute_unl` は  $u_{nl}^\alpha$  を評価する。

```
import irbasis
import numpy

b = irbasis.load('F', 1000)
print("dim: ", b.dim())

# All singular values
print("singular values: ", b.sl())
print("u_0(0.1)", b.ulx(0, 0.1))
print("v_0(0.1)", b.vly(0, 0.1))

print("k-th derivative of u_l(x) and v_l(y)")
for k in range(1,3):
    print(" ", k, b.d_ulx(0, 0.1, k))
    print(" ", k, b.d_vly(0, 0.1, k))

# Compute u_{nl} as a matrix for the first
# 10 non-negative Matsubara frequencies
unl = b.compute_unl(numpy.arange(10))
print("matrix unl ", unl.shape)
```

実行すると、瞬時に以下のような結果が得られるだろう。

```
dim: 72
singular values: [6.93676632e-02
 6.39029304e-02 4.73525981e-02
 3.78765556e-02
 2.69709397e-02 1.97919952e-02 1.38987851e
-02 9.79170114e-03
 6.77586537e-03 4.66563008e-03 3.18095991e
-03 2.15479043e-03
 1.44908097e-03 9.68433898e-04 6.43222579e
-04 4.24765651e-04
 2.78944576e-04 1.82210441e-04 1.18411677e
-04 7.65700139e-05
 4.92756267e-05 3.15628325e-05 2.01254341e
-05 1.27758728e-05
 8.07529293e-06 5.08266091e-06 3.18587477e
-06 1.98887606e-06
 1.23669448e-06 7.65992957e-07 4.72633737e
-07 2.90529576e-07
 1.77929561e-07 1.08573268e-07 6.60143404e
-08 3.99959952e-08
 2.41478607e-08 1.45293028e-08 8.71232105e
-09 5.20671801e-09
 3.10136291e-09 1.84126342e-09 1.08960675e
-09 6.42730925e-10
 3.77927545e-10 2.21524423e-10 1.29443906e
-10 7.54051848e-11
 4.37917794e-11 2.53552246e-11 1.46364946e
-11 8.42388202e-12
 4.83396492e-12 2.76580014e-12 1.57787966e
-12 8.97577822e-13
 5.09125136e-13 2.87965124e-13 1.62415181e
-13 9.13467044e-14
 5.12326226e-14 2.86546680e-14 1.59825721e
```

```
-14 8.89012035e-15
4.93157043e-15 2.72826291e-15 1.50527941e
-15 8.28293591e-16
4.54562830e-16 2.48800912e-16 1.35820168e
-16 7.39497064e-17]
u_0(0.1) 0.27401896348952326
v_0(0.1) 0.7859154340971242
k-th derivative of u_l(x) and v_l(y)
 1 0.04705147674680368
 1 -5.044715458614918
 2 0.4853426135802095
 2 76.97205690788114
unl (10, 72)
```

irbasis は、 $x, y, n$  のベクトルデータに対する基底関数を一回の関数呼び出しで一度に計算できる (numpy の broadcasting に対応, 下記の例参照)。

#### 5.4 2次摂動の計算例

では、簡単な例として、2次摂動の範囲で、2次元ハバード模型の自己エネルギーを計算してみよう。以下のコードは Jupyter notebook として公開されている [22]。

まずは、 $\beta = 10^3$ ,  $\omega_{\max} = 10^2$  ( $\Lambda = 10^5$ ) に対して、計算に必要なデータ、サンプリング点の位置と、そこでの基底関数の値を計算する。特に、式 (13), (14) に現れる  $\hat{\mathbf{F}}_{\alpha}^{+}$ ,  $\mathbf{F}_{\alpha}^{+}$  を計算する。なお、サンプリング点の生成関数 (sampling\_points\_x, sampling\_points\_matsubara) には、引数として展開に使う基底関数の最高次数 (基底関数の数 -1) を与える。今回のサンプルでは、収録されている全ての (137 個) の基底を展開に使う。

下のコードの実行後、変数 invFtau, invhatF がそれぞれ  $\mathbf{F}_{\alpha}^{+}$ ,  $\hat{\mathbf{F}}_{\alpha}^{+}$  の 2次元配列である\*8。

```
import irbasis
import numpy as np

beta = 20.0
Lambda = 10000.0
wmax = Lambda/beta
print("wmax", wmax)

b = irbasis.load('F', wmax * beta)
dim = b.dim()

# Sampling points in x
```

\*8 これらのデータをファイルに書き出す場合には、無次元化した表示を使う方が便利である。

```

sp_x = b.sampling_points_x(whichl=dim-1)
print(sp_x)

# Sampling frequencies in n
sp_n = b.sampling_points_matsubara(whichl=dim-1)
print(sp_n)

all_ls = np.arange(dim)

# Compute F and pseudo inverse
# for imaginary time
Ftau = np.sqrt(2/beta) * \
    b.ulx(all_ls[:, None], sp_x[None,:]).T
invFtau = np.linalg.pinv(Ftau)

# Compute F and pseudo inverse
# for imaginary frequency
hatF = np.sqrt(beta) * b.compute_unl(sp_n)
invhatF = np.linalg.pinv(hatF)
    
```

次に、ベアなグリーン関数をバンド分散に

$$\epsilon(k) = -2(\cos k_1 + \cos k_2) \quad (30)$$

に対して計算する。ここで、グリーン関数

$$G(k, i\omega) = \frac{1}{i\omega - \epsilon(k)} \quad (31)$$

は、サンプリング周波数点上のみで計算される。

```

# Parameters
nk_lin = 64
U, kps = 2.0, np.array([nk_lin, nk_lin])
nw = len(sp_n)
ntau = len(sp_x)

# Generate k mesh and non-interacting band energies
nk = np.prod(kps)
kgrid = [2*np.pi*np.arange(kp)/kp for kp in kps]
k1, k2 = np.meshgrid(*kgrid, indexing='ij')
ek = -2*(np.cos(k1) + np.cos(k2))
iw = 1j*np.pi*(2*sp_n+1)/beta

# G(iw, k): (nw, nk)
gkf = 1.0 / (iw[:,None] - ek.ravel()[None,:])
    
```

この段階で、 $gkf$  は、サンプリング周波数点  $\times$  波数の 2 次元配列である。さて、2 次摂動の自己エネルギーは、虚時間、実空間で

$$\Sigma(\tau, r) = U^2 G(\tau, r) G(\beta - \tau, r) \quad (32)$$

のように対角な形を取る ( $r$  は実空間の座標)。従って、サンプリング周波数  $\rightarrow$  IR 基底  $\rightarrow$  虚時間での

サンプリング点への変換、波数空間から実空間へのフーリエ変換を続けて実行して、 $G(\tau, r)$  を求めた後、 $\Sigma(\tau, r)$  を計算する。

```

# G(l, k): (L, nk)
gk1 = np.dot(invhatF, gkf)

# G(tau, k): (ntau, nk)
gkt = np.dot(Ftau, gk1)

# G(tau, r): (ntau, nk)
grt = fftn(gkt.reshape(ntau, *kps), axes=(1,2)).\
    reshape(ntau, nk)

# Sigma(tau, r): (ntau, nk)
srt = U*U*grt*grt*grt[:, :-1, :]
    
```

この段階で、計算結果  $\Sigma(\tau, r)$  は、 $srt$  に 2 次元配列として保存されている。最後に、IR 基底を経由して、虚周波数におけるサンプリング点上へ変換した後、波数空間にフーリエ変換すれば良い。

```

# Sigma(l, r): (L, nk)
srl = np.dot(invFtau, srt)

# Sigma(iw, r): (nw, nk)
srf = np.dot(hatF, srl)

# Sigma(iw, k): (nw, kps[0], kps[1])
srf = srf.reshape(len(sp_n), *kps)
skf = ifftn(srf, axes=(1,2))/nk**2
    
```

最後に、2 次摂動の自己エネルギーを図 8 にプロットする。スパースサンプリング法を使うことで、2 次摂動計算が少数のサンプリング点上で閉じる。

## 6 まとめと今後の展望

本稿では、グリーン関数のコンパクトな表現やスパースサンプリング法に基づく高精度かつ高速な数値計算法を紹介した。IR 基底を使うことで、データサイズは逆温度に対して対数的にしか増大しない。さらに、スパースサンプリング法を使うことで、数値データのコンパクトさを保ったまま、高速にダイアグラム方程式を解くことが可能になる。これらの手法は、広いバンド幅や低温という異なるエネルギースケールを同時に扱う必要がある第一原理計算において極めて有用である。その例として、 $GW$  近似、Migdal-Eliashberg 理論などに基づく第一原理計算

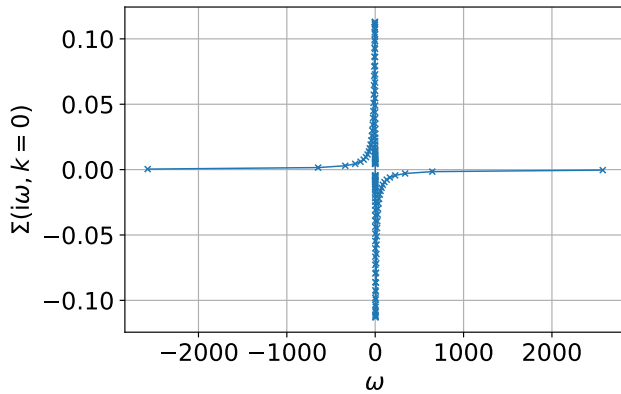


図8 2次元ハバード模型に対して計算した自己エネルギー ( $k=0$ ). スパースモデリング法を使うことで, 少数 ( $\simeq 140$  個) のサンプル点上で計算が閉じる.

の結果を紹介した. さらに, 本稿では, これらの手法を簡単に使える数値計算ライブラリ `irbasis` の概要とサンプルコードを紹介した. 本稿が様々な分野での応用のきっかけになることを願う. もし, 読者がデータサイズ・計算時間の問題に困っていたら, 是非相談して欲しい.

最後に, 今後の展望を述べる. 研究のフロンティアは, 2粒子量にすでに移っている. バートックス補正を考慮したダイアグラム計算や, 動的平均場理論による動的感受率計算, 非従来型超伝導を扱うことができる動的平均場理論の非局所拡張 (デュアルフェルミオン法 [23, 24] や動的バートックス近似法 [25] など) などには, 2粒子量 (2粒子グリーン関数や4点バートックス関数) の効率的な数値的取り扱いが必須である. これらの2粒子量は3つの虚周波数に依存するだけではなく, 軌道・スピン・波数に依存するため, その数値的取扱は難しい.

これらの問題に対して, 最近突破口が開かれた. 近年, 品岡, 大槻, 吉見と共同研究者は, IR基底を基にした「過剰基底」を使うことで, 2粒子量の複雑な周波数依存性をコンパクトに表現可能であることを示した [4]. また, スパースサンプリング法も2粒子量に拡張された [26]. さらに, 品岡とウィーン工科大の共同研究者は, 過剰基底とスパースサンプリング法を組み合わせることで, ベーテ・サルペータ方程式がIR基底上で解けることをごく最近示した [27]. ま

た, 2粒子量の数値データを, 周波数・軌道・スピンに対する多次元データと見なし, テンソル分解を適用することで, データ量を軽減する試みも始まっている [26]. ある種の例では, テラバイト級の2粒子グリーン関数の数値データを1メガバイト以下へ, 桁違いに圧縮できることが示されている [26]. このような基礎技術の開発が技術的な飛躍につながり, 高精度な第一原理計算に基づく強相関化合物の物性の理解につながることを期待している. また, 一連の手法は物性に限らない汎用的なもので, 高エネルギー, 核物理などへの応用も面白い課題である.

## 謝辞

本稿で紹介した研究の共同研究者に感謝を致します. スパースモデリングを量子多体計算へ応用しようとする, 大関真之先生との共同研究が, IR基底の発見につながりました. また, スパースサンプリング法の開発には, ミシガン大学の Gull グループとの共同研究が非常に有益でした. 本研究は以下に列挙する科学研究費補助金の支援を受けて行われました: No. 15H05885 (J-Physics), No. 16H01064 (J-Physics), No. 18H04301 (J-Physics), No. 18H01158, No. 16K17735, No. 19K14654.

## 参考文献

- [1] 大槻純也, 大関真之, 品岡寛, and 吉見一慶: 固体物理 **53** (2018) 173.
- [2] N. Chikano, K. Yoshimi, J. Otsuki, and H. Shinaoka: Computer Physics Communications **240** (2018) 181.
- [3] H. Shinaoka, J. Otsuki, M. Ohzeki, and K. Yoshimi: Physical Review B **96** (2017) 035147.
- [4] H. Shinaoka, J. Otsuki, K. Haule, M. Wallerberger, E. Gull, K. Yoshimi, and M. Ohzeki: Physical Review B **97** (2018) 205111.
- [5] M. Wallerberger, private communication.
- [6] L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, and O. Parcollet: Physical Review B **84** (2011) 075145.
- [7] J. Otsuki, M. Ohzeki, H. Shinaoka, and K. Yoshimi: Physical Review E **95** (2017) 061302(R).
- [8] J. Otsuki, M. Ohzeki, H. Shinaoka, and K. Yoshimi: Journal of the Physical Society of Japan **89** (2020) 012001.
- [9] J. Li, M. Wallerberger, N. Chikano, C.-N. Yeh,

- E. Gull, and H. Shinaoka: Physical Review B **101** (2020) 035144.
- [10] M. Kaltak and G. Kresse: Physical Review B **101** (2020) 205145.
- [11] T. Wang, T. Nomoto, Y. Nomura, H. Shinaoka, J. Otsuki, T. Koretsune, and R. Arita: Physical Review B **102** (2020) 134503.
- [12] A. I. Liechtenstein, M. I. Katsnelson, and V. A. Gubanov: Journal of Physics F: Metal Physics **14** (1984) L125.
- [13] T. Nomoto, T. Koretsune, and R. Arita: Physical Review B **102** (2020) 014444.
- [14] T. Nomoto, T. Koretsune, and R. Arita: Physical Review Letters **125** (2020) 117204.
- [15] Y. Nomura, T. Nomoto, M. Hirayama, and R. Arita: Physical Review Research **2** (2020) 043144.
- [16] N. Witt, E. G. C. P. van Loon, T. Nomoto, R. Arita, and T. Wehling, arXiv:2012.04562.
- [17] S. Iskakov, C.-N. Yeh, E. Gull, and D. Zgid: Physical Review B **102** (2020) 085105.
- [18] Y. Nagai and H. Shinaoka: Journal of the Physical Society of Japan **88** (2019) 064004 .
- [19] H. Shinaoka and Y. Nagai: Physical Review B **103** (2021) 045120.
- [20] N. Chikano, J. Otsuki, and H. Shinaoka: Physical Review B **98** (2018) 035104.
- [21] [https://nbviewer.jupyter.org/github/SpM-lab/ir\\_kotaibutsuri/blob/82d34fe0bd7f5313fbdfa87a38983aadba2d85cb/sample.ipynb/](https://nbviewer.jupyter.org/github/SpM-lab/ir_kotaibutsuri/blob/82d34fe0bd7f5313fbdfa87a38983aadba2d85cb/sample.ipynb/).
- [22] [https://nbviewer.jupyter.org/github/SpM-lab/ir\\_kotaibutsuri/blob/82d34fe0bd7f5313fbdfa87a38983aadba2d85cb/second\\_order\\_perturbation.ipynb/](https://nbviewer.jupyter.org/github/SpM-lab/ir_kotaibutsuri/blob/82d34fe0bd7f5313fbdfa87a38983aadba2d85cb/second_order_perturbation.ipynb/).
- [23] A. N. Rubtsov, M. I. Katsnelson, and A. I. Liechtenstein: Physical Review B **77** (2008) 033101 .
- [24] 大槻純也 and 楠瀬博明: 固体物理 **51** (2016) 223.
- [25] A. Toschi, A. Katanin, and K. Held: Physical Review B **75** (2007) 045118 .
- [26] H. Shinaoka, D. Geffroy, M. Wallerberger, J. Otsuki, K. Yoshimi, E. Gull, and J. Kuneš: SciPost Physics **8** (2020) 012 .
- [27] M. Wallerberger, H. Shinaoka and A. Kauch, arXiv:2012.05557v1.