確率分布推定のための逆自己畳み込みアルゴリズム

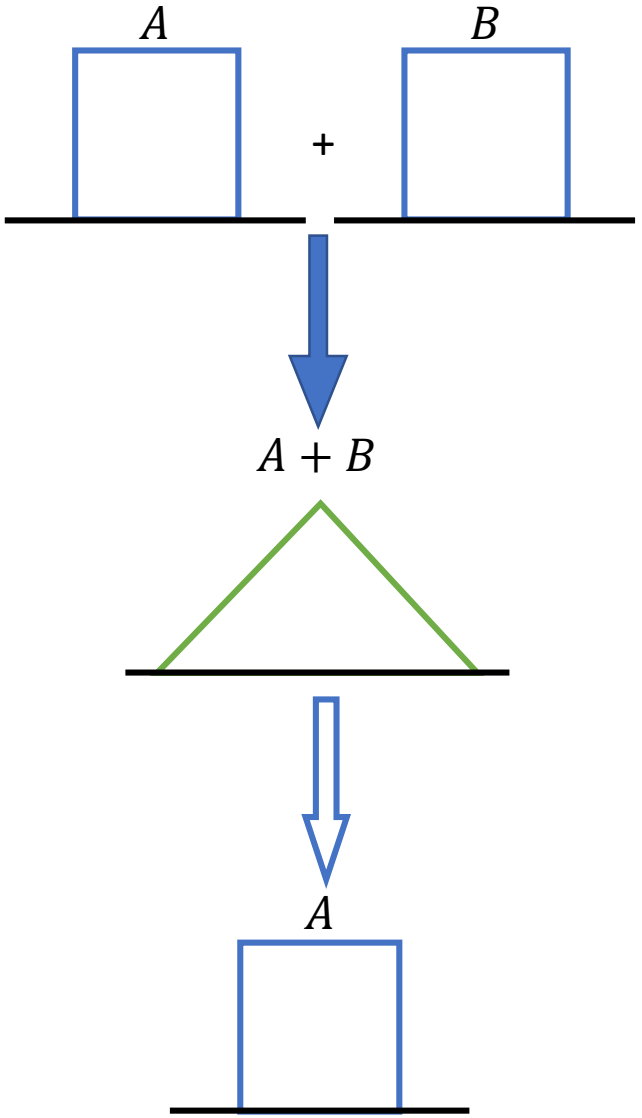
山村真由子（大阪工業大学）
福井聡史（大阪工業大学）
尾張優哉（大阪工業大学）
重弘裕二（大阪工業大学）

動機

- 組合せ最適化手法について考察
  - 解の評価値の分布を推定
    - 同じ分布をもつ確率変数の和の分布

自分自身との畳み込みの逆演算
（**逆自己畳み込み**とよぶ）

元の確率変数の分布

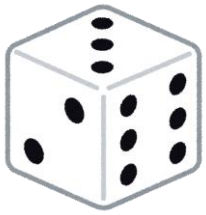


例）２つのサイコロの目の和

A, B はたがいに影響しない（独立）

どの目が出る確率も $\frac{1}{6}$

サイコロ A



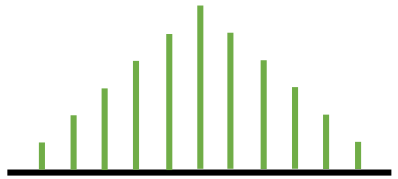
サイコロ B

２つのサイコロの目の和

$(A, B) = (1, 1)$
→ 確率は $\frac{1}{36}$

$(A, B) = (1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)$
→ 確率は $\frac{6}{36} = \frac{1}{6}$

拡大

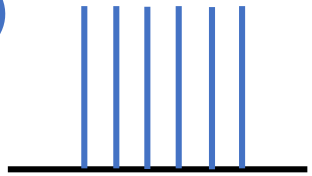


サイコロ $A + B$

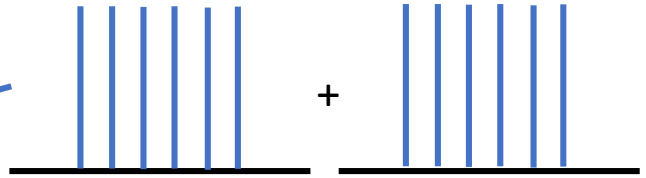
逆自己畳み込み

逆演算

サイコロ A



サイコロ A　サイコロ B



自己畳み込み

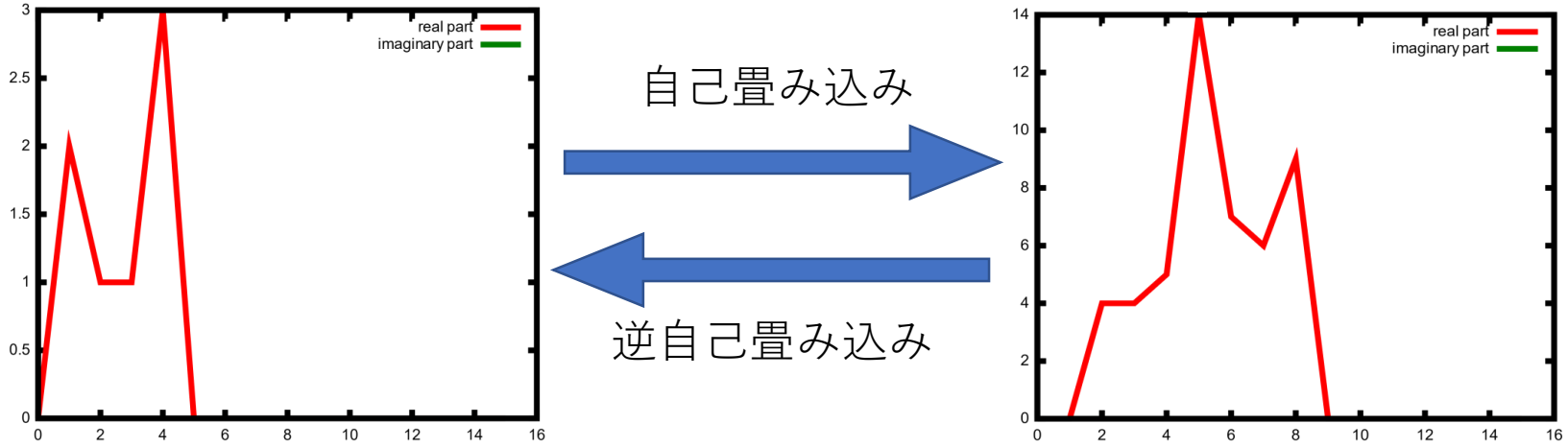
研究内容

- 逆自己畳み込みアルゴリズム
  - **自己畳み込み**は**フーリエ変換の二乗**に対応

  ⇒フーリエ変換の平方根を求めるアルゴリズム
    - 複素数の平方根は２つあり、一意に定まらない

    ⇒フーリエ変換の平滑化

      外挿に基づく平方根の推定



畳み込みとフーリエ変換

- データ列

  $f : f_0, f_1, \cdots$

- 畳み込み

  $f * g = c,$ $\quad c_n = \sum f_{(n-m)}\, g_m$

- 離散フーリエ変換

  $\mathcal{F}[f] = F,$ $\quad F_k = \sum W^{kn} f_n$

  $(W = \exp(-j\frac{2\pi}{N}))$ $\quad N$: データ数

- 逆離散フーリエ変換

  $\mathcal{F}^{-1}[f] = f,$ $\quad f_n = \sum W^{-kn} F_k$

データ列は小文字で表す

各要素を小文字に下付きの数字で表す

f と g の畳み込みをアスタリスクで表す

定義式

離散フーリエ変換を$\mathcal{F}$で表す

定義式

逆離散フーリエ変換を $\mathcal{F}^{-1}$ で表す

定義式

畳み込みのフーリエ変換

- 畳み込みのフーリエ変換はフーリエ変換の積

$$\mathcal{F}[f * g] = \mathcal{F}[f]\mathcal{F}[g]$$

- 従来の逆畳み込み
  - ボケ画像の復元等に適用される

$$f,\ f * g \rightarrow \mathcal{F}[f],\ \mathcal{F}[f * g]$$
$$\rightarrow \mathcal{F}[g] = \frac{\mathcal{F}[f*g]}{\mathcal{F}[f]}$$
$$\rightarrow g = \mathcal{F}^{-1}\left[\mathcal{F}[g]\right]$$

f　レンズの影響
g　もとの画像
$f * g$　ボケ画像

$f, f * g$ が
わかったら

逆自己畳み込み

- 逆畳み込みと同じようにできそうなのですが…

1. フーリエ変換を求める

   $f * f \rightarrow \mathcal{F}[f * f] = (\mathcal{F}[f])^2$

2. 平方根を求める

   $(\mathcal{F}[f])^2 \rightarrow \mathcal{F}[f]$

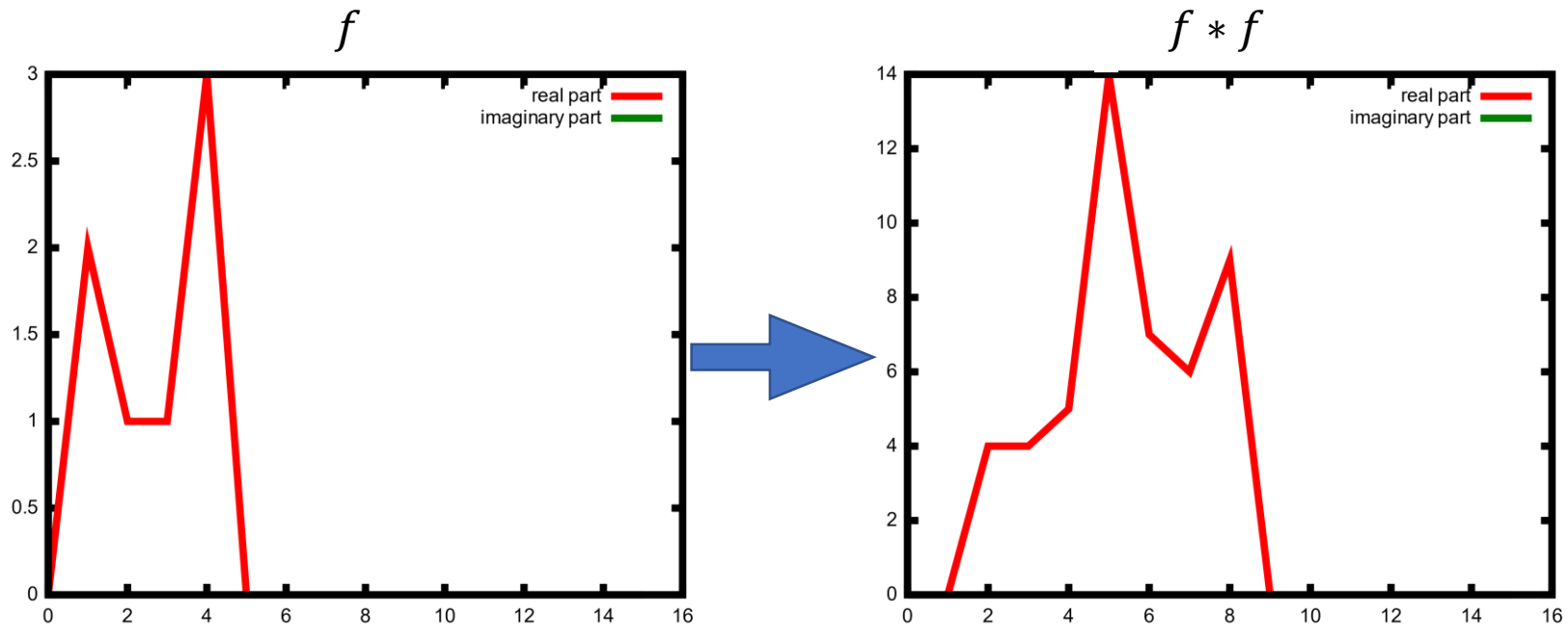
3. 逆フーリエ変換を求める

   $\mathcal{F}\,[f] \rightarrow f$

どうやって
$(\mathcal{F}[f])^2$ の平方根を
求めればいいのか

自己畳み込みの例

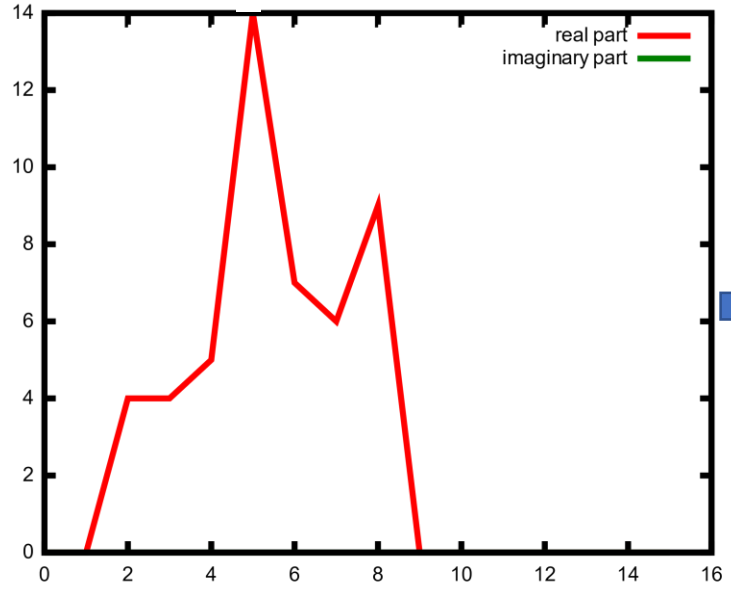
- 横軸がデータ番号
- 縦軸が大きさ
- 隣り合うデータを線でむすんでいる



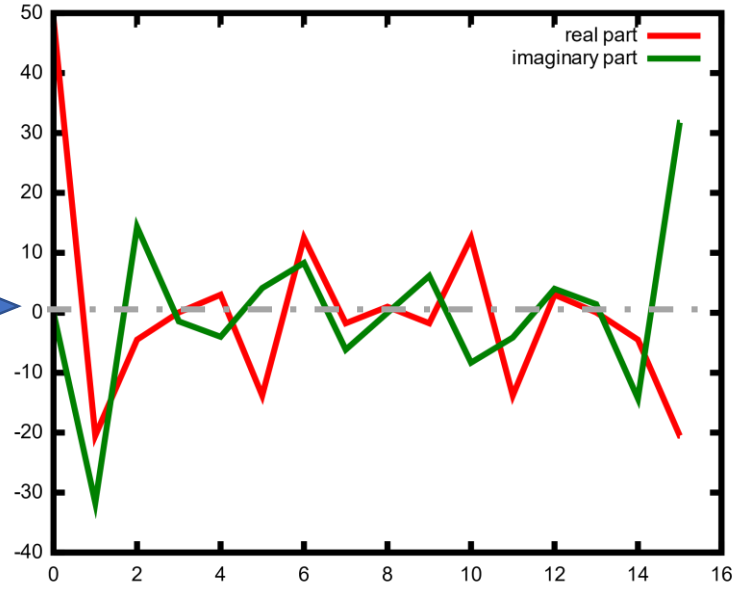
自己畳み込みのフーリエ変換の例

- 赤色が実部、緑色が虚部
- 正負に振動

$f * f$

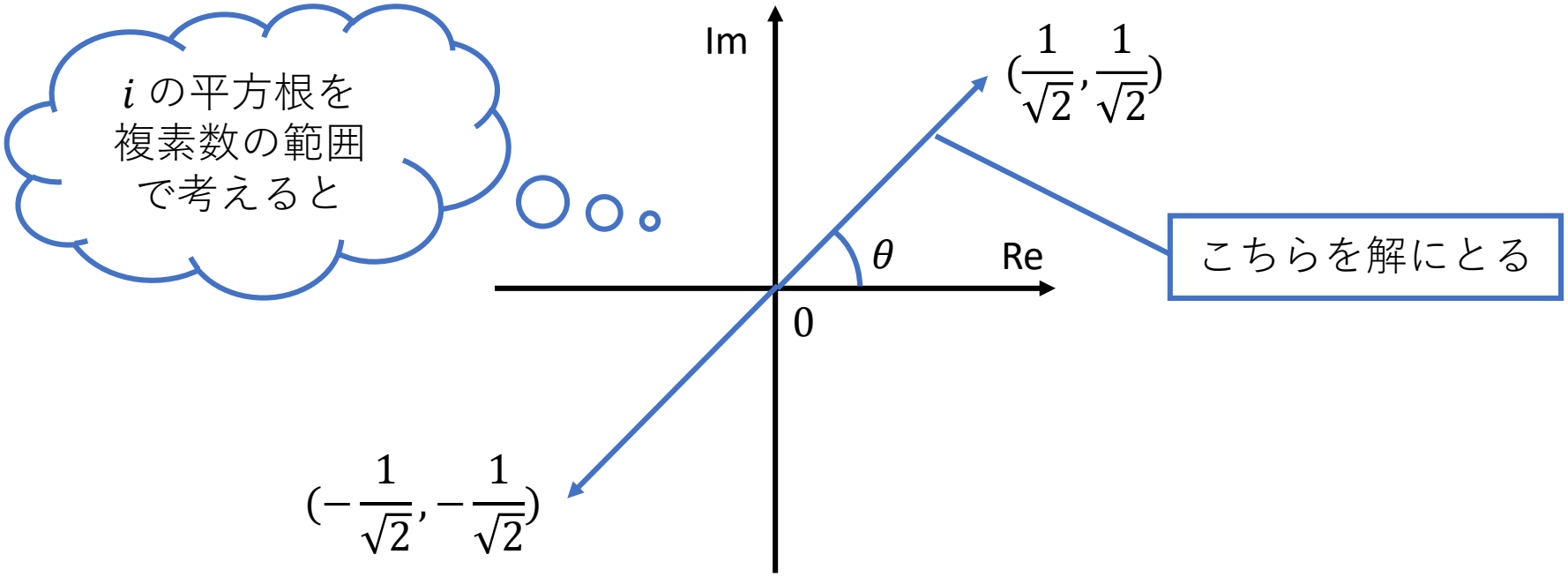


$F(f * f) = (F[f])^2$



複素数の平方根

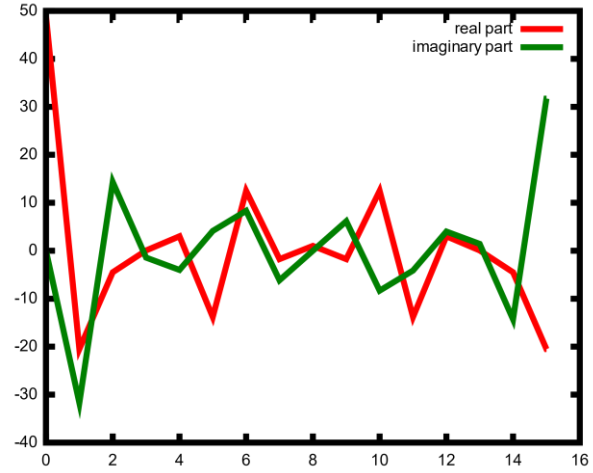
- 複素数の平方根は２つある
- プログラミング言語のsqrt関数では基本的に
  $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$ の範囲の値を返す　　θ: 偏角



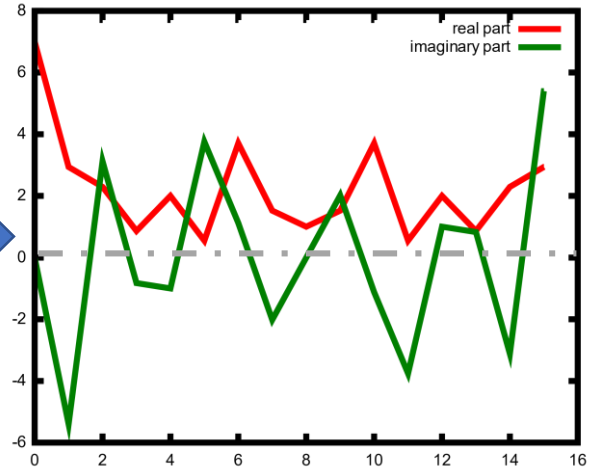
平方根を単にsqrt関数で求めた場合

- プログラミング言語のsqrt関数では、２つの平方根のうち、実部が非負の平方根が求まる
- 元のフーリエ変換を求めるのは無理

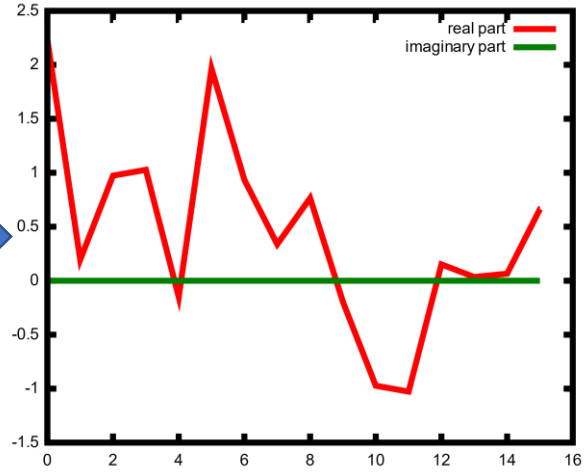
f の自己畳み込みのフーリエ変換だったとする
$(\mathcal{F}[f])^2$



sqrt関数で平方根を求めた結果



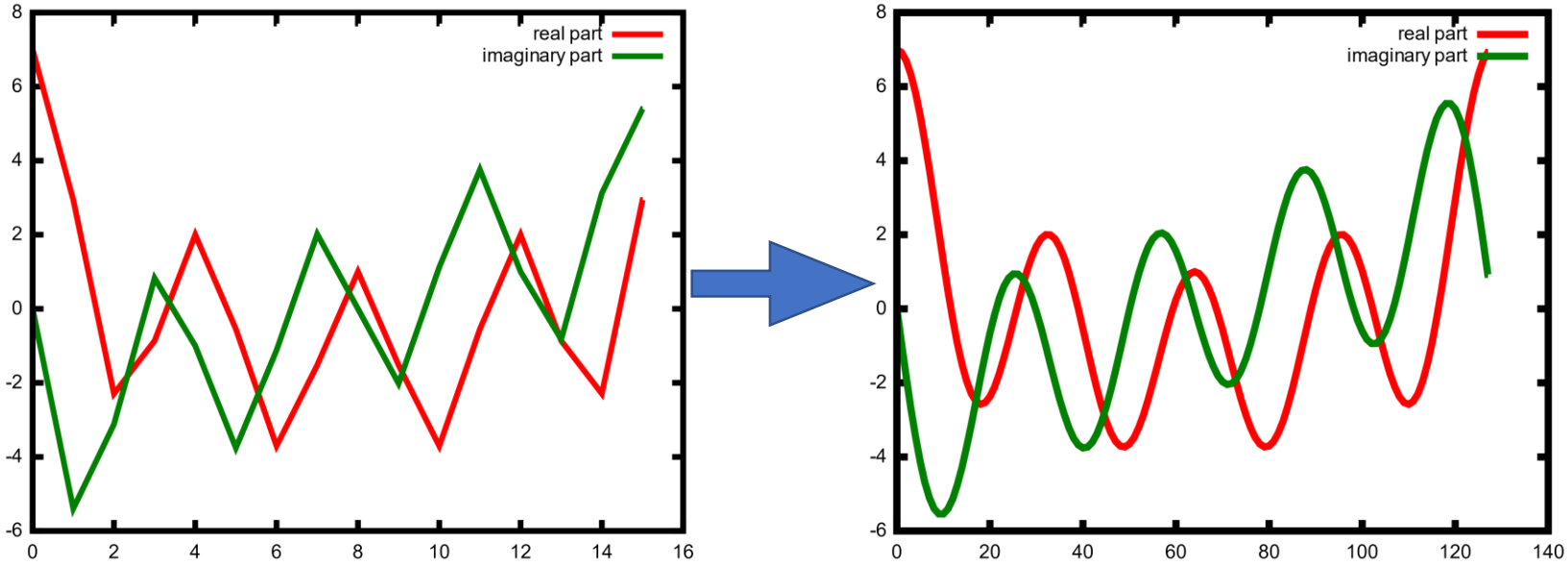
さらに逆フーリエ変換をすると f が求まるはずですが…



解決案

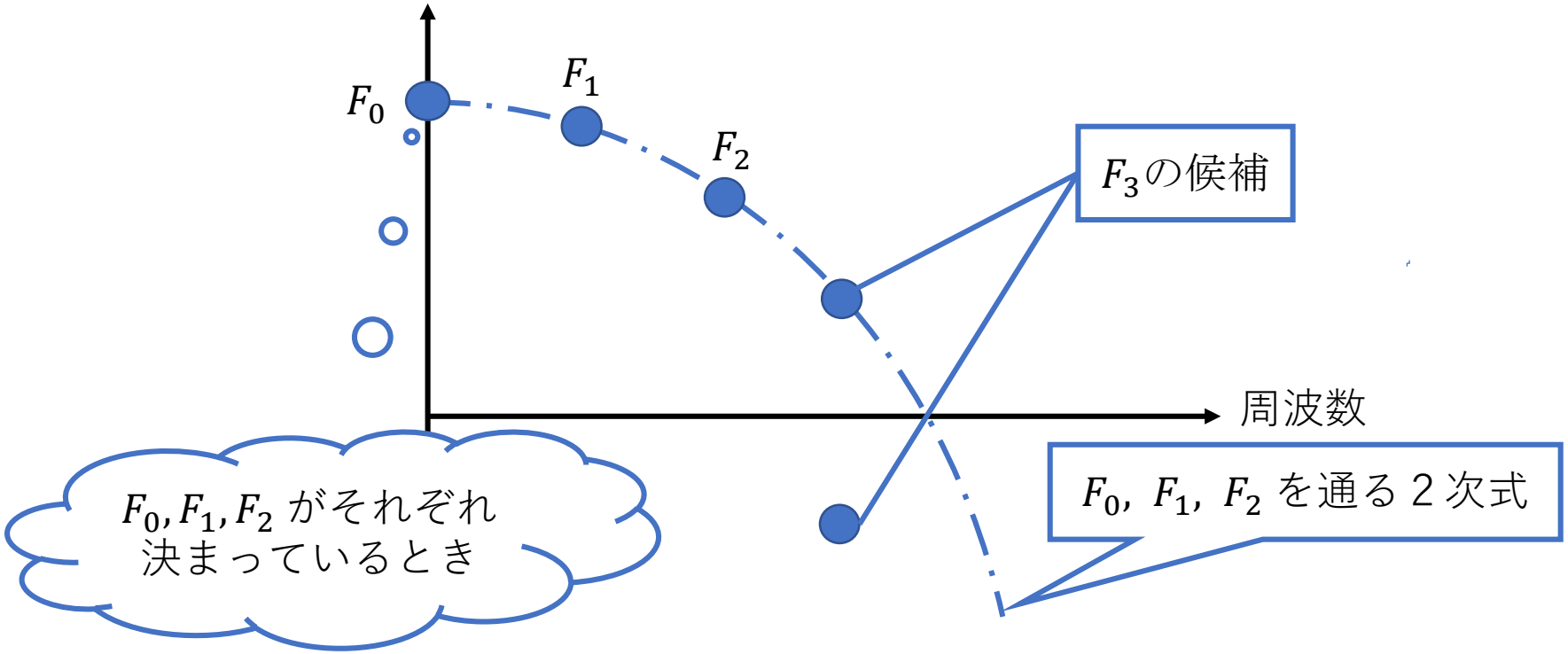
どうやって？

- フーリエ変換を平滑化
  - 滑らかにつながっていれば、２つの平方根の片方を選定できる
    - 近似に基づく外挿



近似に基づく外挿

- フーリエ変換結果のF_0（直流成分）は正のはず
  $F_0 = f_i$　　f_i は確率
- データ数点を通る多項式を考える



平滑化

- フーリエ変換の式の各項の寄与

$$F_k = \sum W^{kn} f_n \qquad (W = \exp(-j\frac{2\pi}{N}))$$

k が 1 増えると

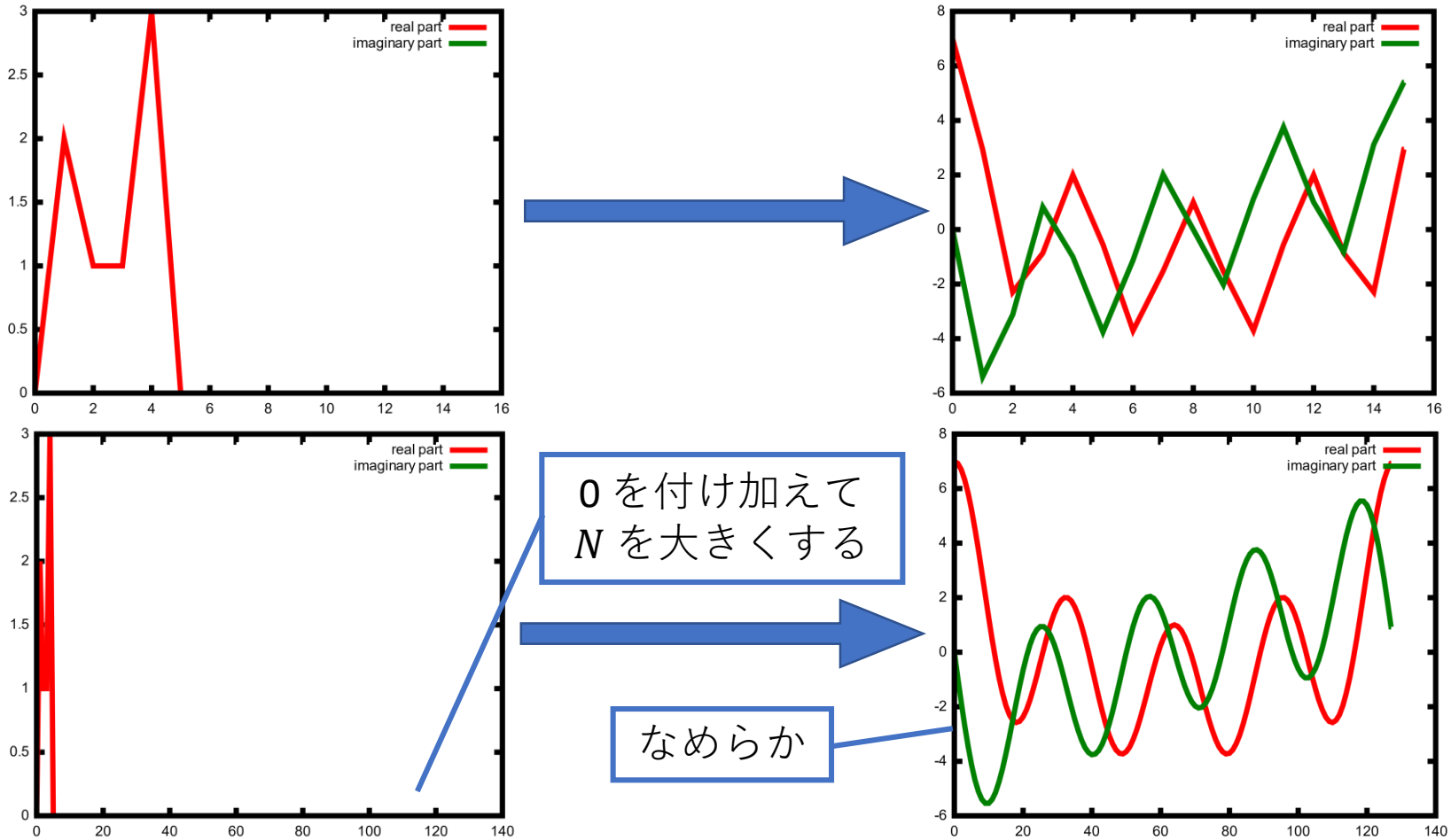
$$W^{(k+1)n} f_n - W^{kn} f_n = (W^n - 1) W^{kn} f_n$$

- 変化量が小さければ F はなめらか
  - N が大きいとき $W \cong 1$
    - n が小さいとき $W^n \cong 1$
    - n が小さくないとき $f_n = 0$
- 中央部のf_nが0なら、Fはなめらか

→補完、推移

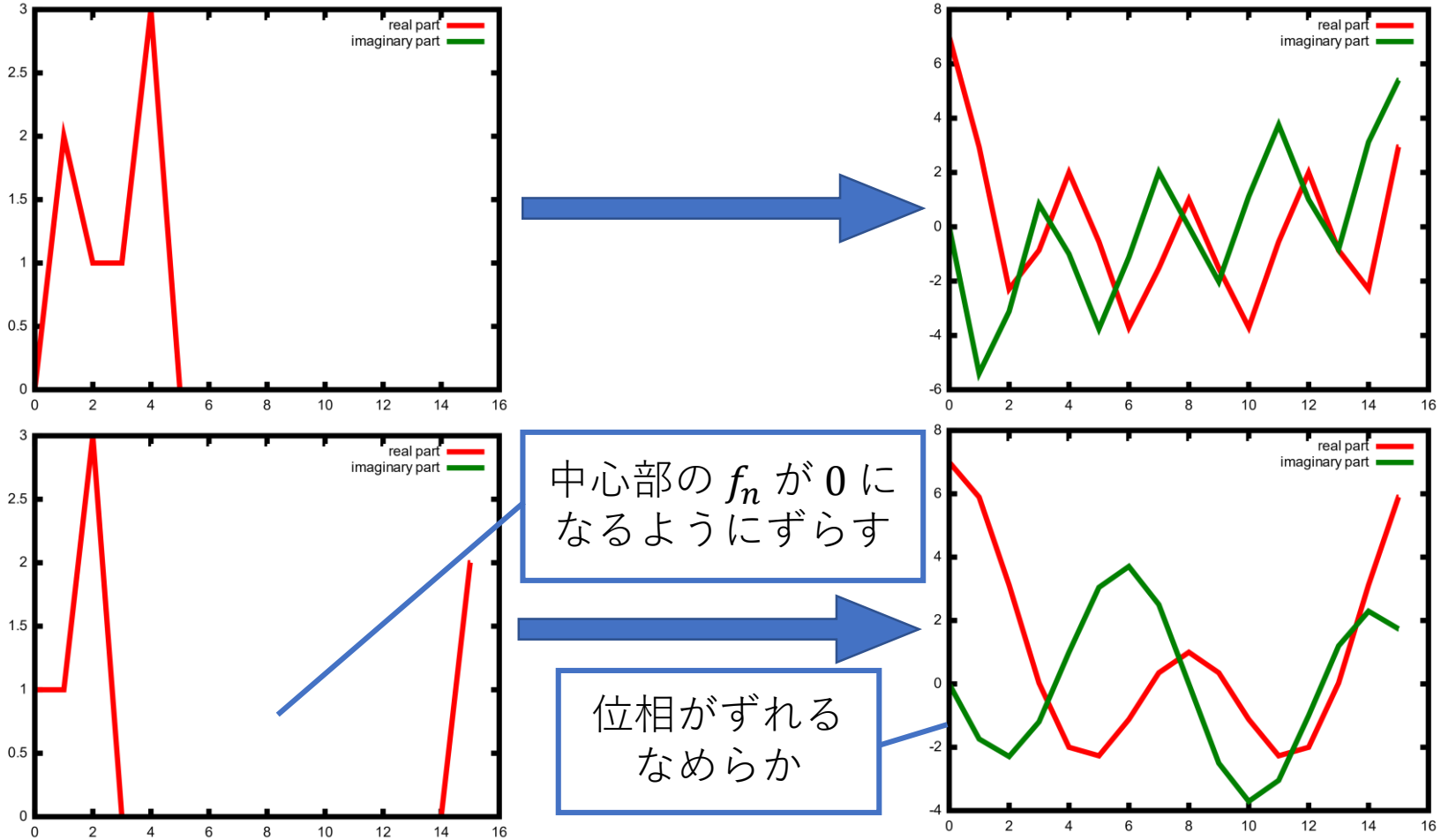
r倍補完

- 補完 ：f に 0 を加えてデータ数を r 倍にする



d推移

- 推移：f を d だけずらす

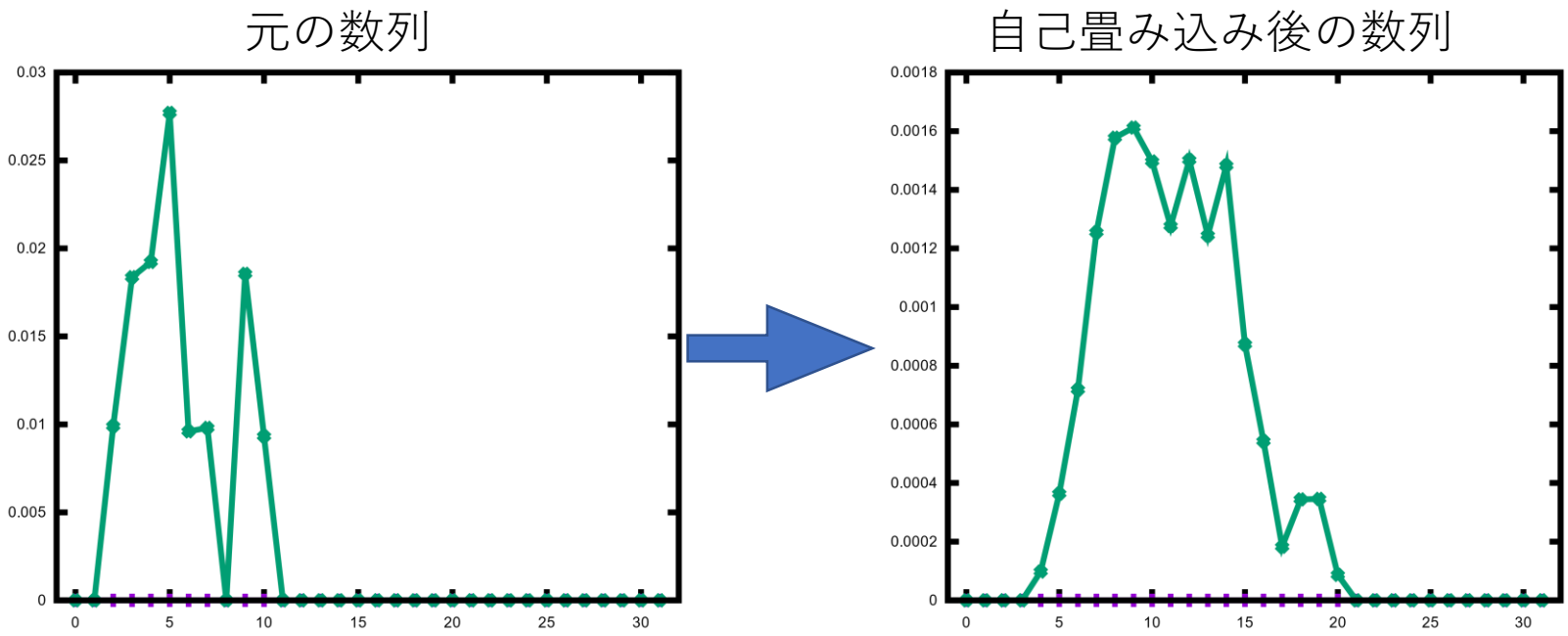


逆自己畳み込みアルゴリズム

1. $f * f$が与えられる
2. 補完と推移
3. フーリエ変換
4. 近似に基づく外挿により平方根を求める
5. 2,3の逆変換

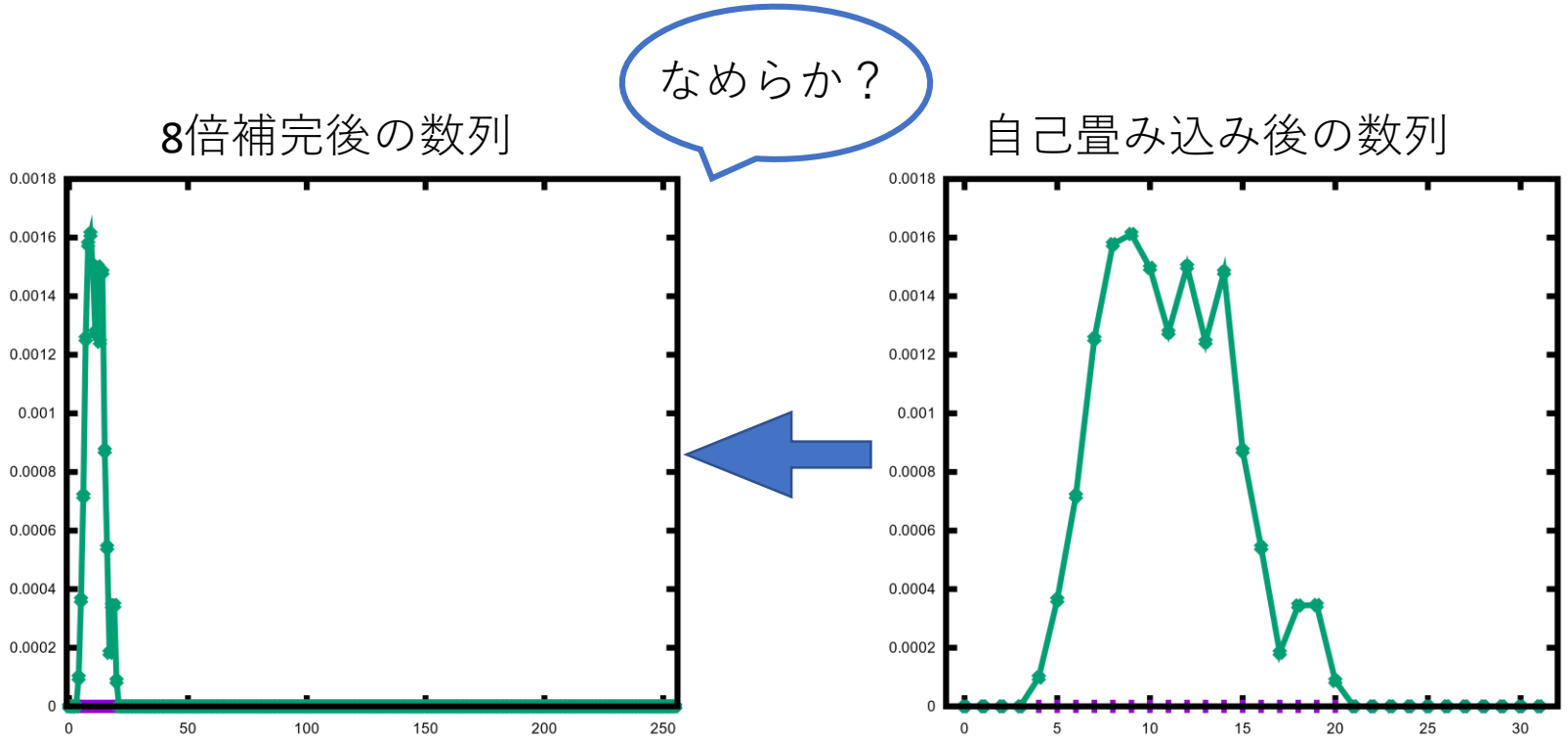
適用例

- 緑色が実部、紫色が虚部
- データ数　$N = 32$



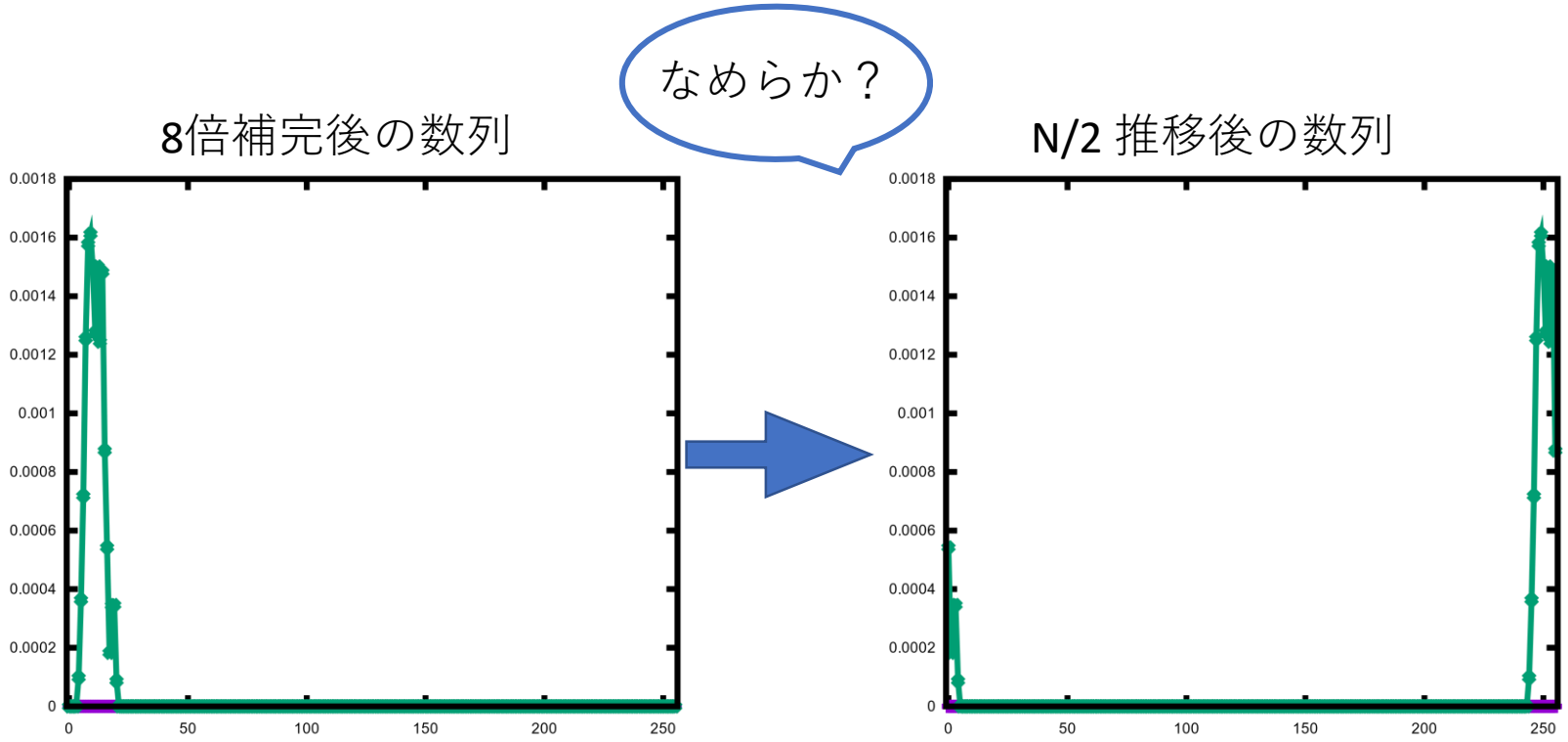
r 倍補完の結果

- 自己畳み込み後の数列に 8 倍補完を行った
- r 倍補完後のデータ数 $N = 256$



$N/2$ 推移の結果

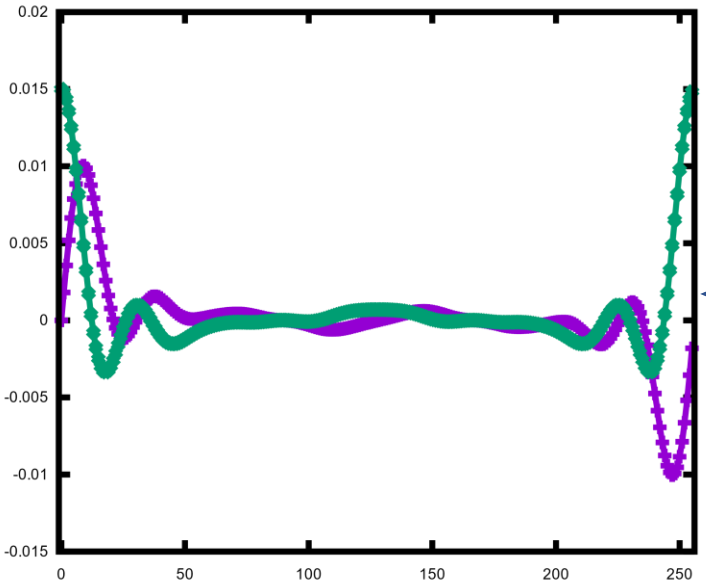
- 8 倍補完後の数列に $N/2$ 推移を行った



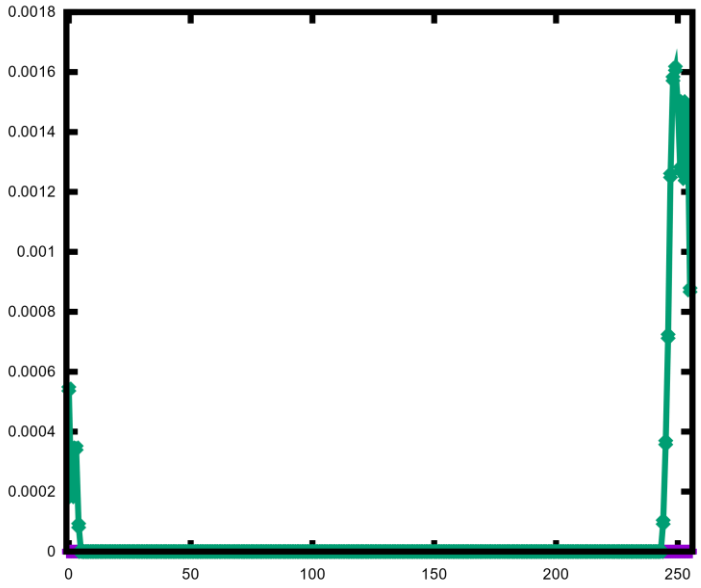
離散フーリエ変換の結果

- 離散フーリエ変換を行った
- 補完と推移により、**なめらか**になっている

離散フーリエ変換後



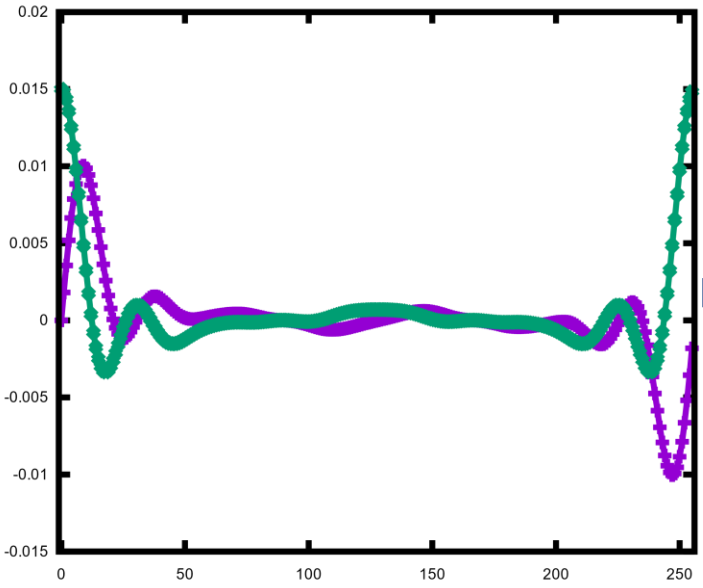
N/2 推移後の数列



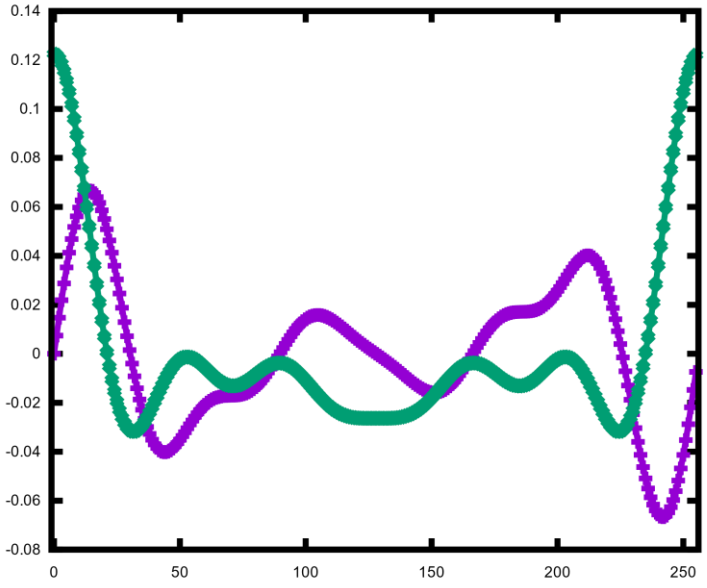
平方根を求めた結果

- 離散フーリエ変換の平方根を求めた
- ４次の多項式で値の外挿を行った

離散フーリエ変換後

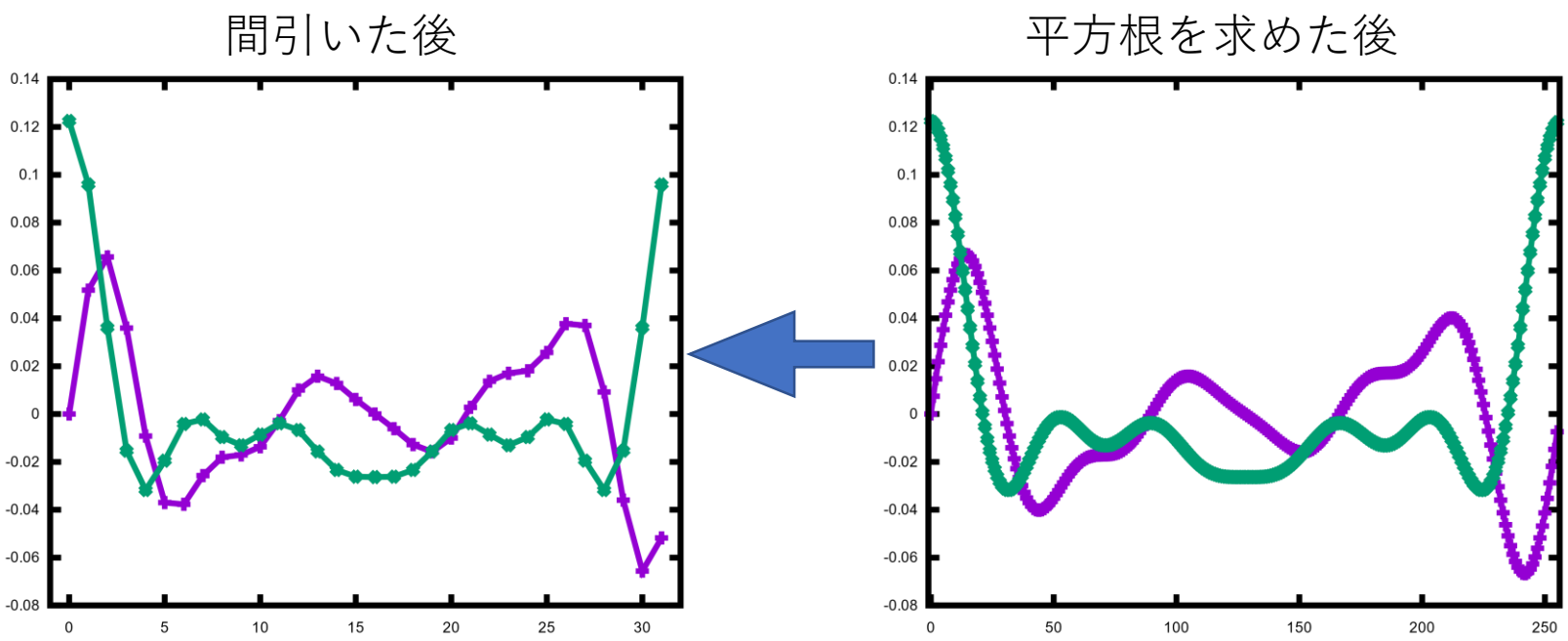


平方根を求めた後



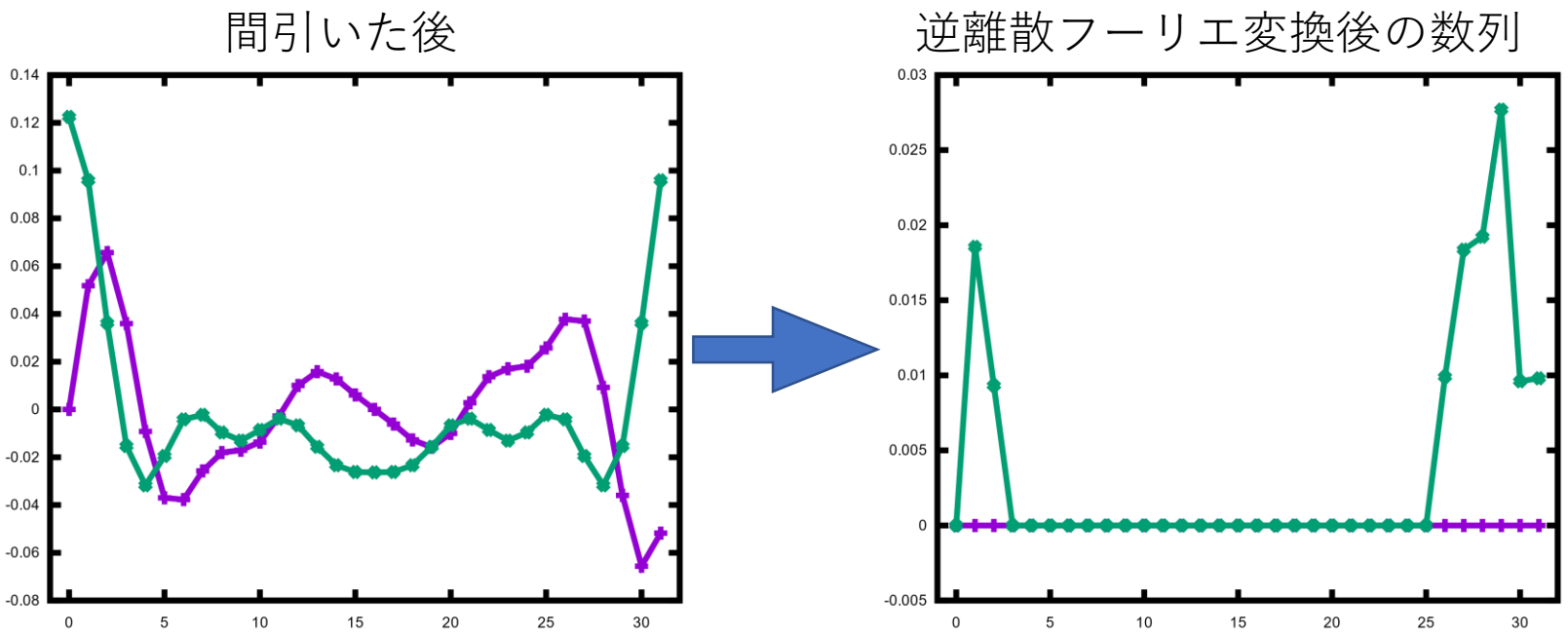
間引いた結果

- 平方根をとった後の数列を間引いた
- データ数は $N = 32$



逆離散フーリエ変換の結果

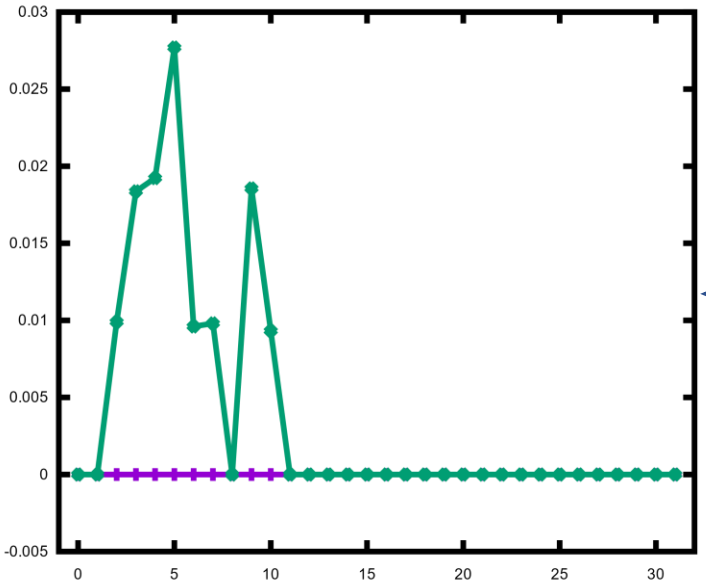
- 逆離散フーリエ変換を行った



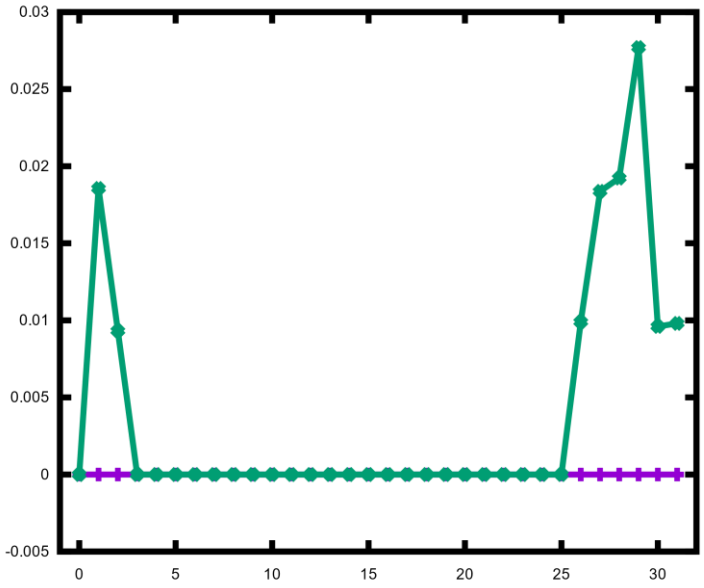
$-N/4$ 推移の結果

- $-N/4$ 推移を行った

$-N/4$ 推移後の数列



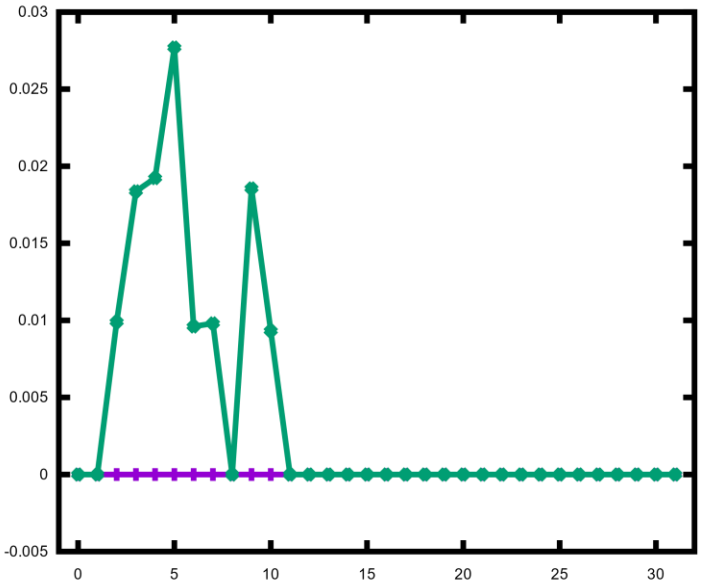
逆離散フーリエ変換後の数列



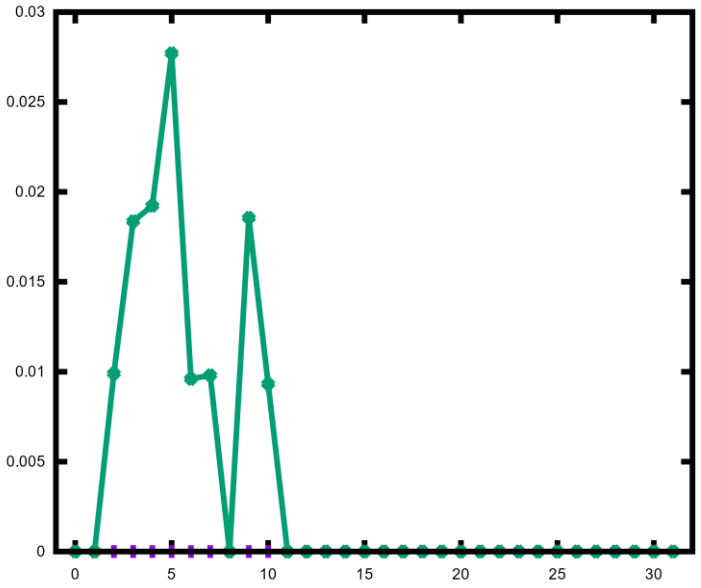
元の数列と処理後の数列

- 同じ数列であることを確認した
- 計算による誤差が含まれるため、処理後の数列のグラフの範囲が違う

処理後の数列



元の数列



まとめ

- 逆自己畳み込みアルゴリズムを開発
- 実際にアルゴリズムを適用し、途中過程の各処理が機能していることが確認できた。