

# FFT(高速フーリエ変換)による周波数解析

## 本ワークシートでの内容

- 音・振動解析において周波数解析により特性を見える化するのがよく行われます。本事例では、時系列の波形について高速フーリエ変換により周波数解析を行う方法が紹介されています。
- 信号処理パッケージ `SignalProcessing` の利用

## 目次

- 時系列波形データを読み込む
  - 時系列データのインポート
  - 時系列データの可視化
- 高速フーリエ変換
  - 高速フーリエ変換の処理
  - フーリエ変換後のデータの可視化

## 主な利用コマンド

コマンド名/演算子名	説明
• <a href="#">ImportMatrix</a>	外部のテキストデータやExcelデータをインポートします。
• <a href="#">plot</a>	2Dのグラフを描画します。
• <a href="#">SignalProcessing:-FFT</a>	高速フーリエ変換を行います。

## 時系列波形データを読み込む

### 時系列データのインポート

外部の時系列データ(Excel ファイル)を Maple にインポートします；

```
[> restart;  
[> data := ImportMatrix("fft_sample.xls", delimiter = ",");
```

処理対象となる部分だけ取り出します；

```
[> sampledata:=data[5 .. 1028, [2, 7]];
```

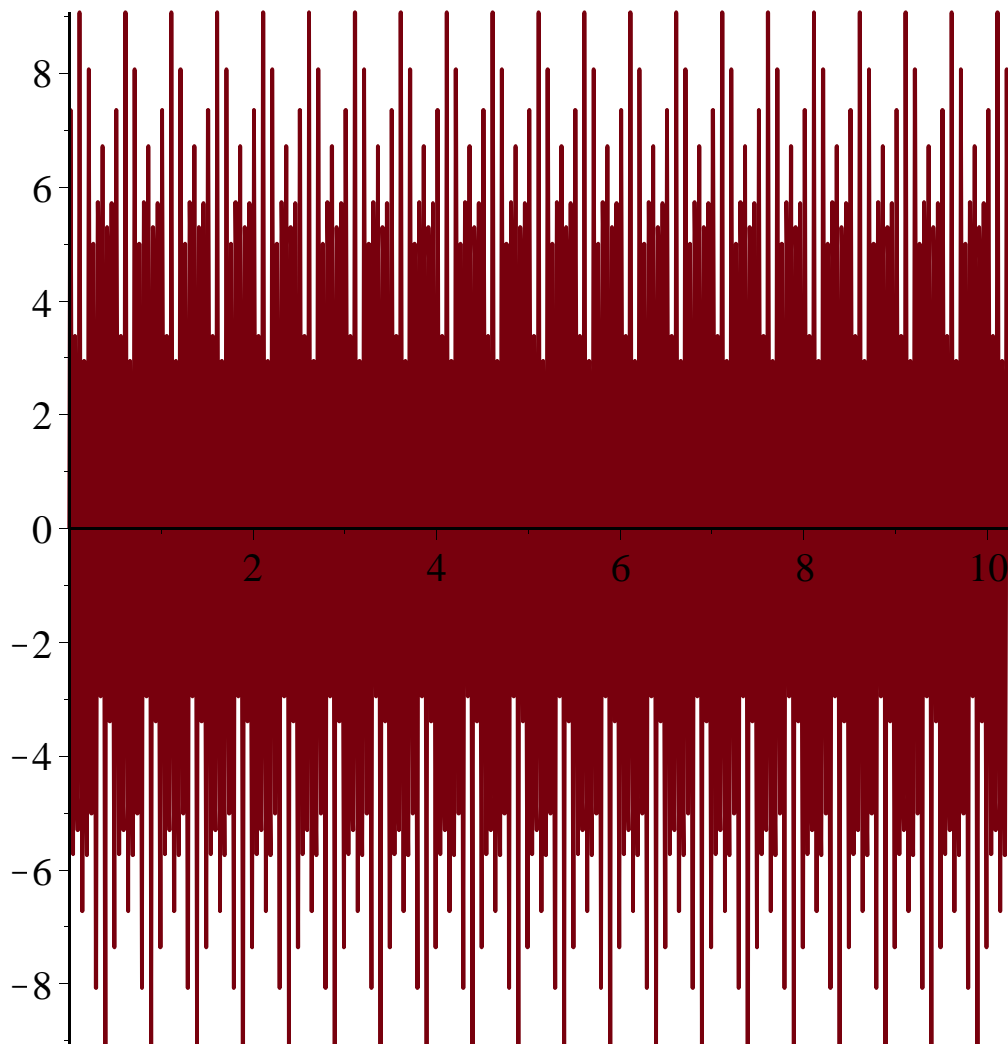
```
sampladata := [ 1024 x 2 Matrix  
                Data Type: anything  
                Storage: rectangular  
                Order: Fortran_order ]
```

(1)

## ▼ 時系列データの可視化

plot コマンドで描画します ;

```
> plot(sampladata, numpoints = 2000);
```



## ▼ 高速フーリエ変換

上記時系列データについて SignalProcessing パッケージの機能を用いてフーリエ変換を行い、変換後のデータについて、周波数・振幅で描画します ;

## ▼ 高速フーリエ変換の処理

SignalProcessing パッケージをロードします ;

```
[> with(SignalProcessing):
```

FFT コマンドを使って波形データをフーリエ変換します。

データの正規化を行う必要がなければ、normalization = none オプションを指定します。

```
[> fq := FFT(sampledata[1 .. -1, 2], normalization = none):
```

サンプリング数を確認します ;

```
[> N:=LinearAlgebra:-Dimension(sampledata)[1];  
N:= 1024
```

(2)

サンプリングの時間刻みを設定します ;

```
[> Ssize := 0.1e-1:
```

周波数を下式で計算します ;

```
[> fqX := [seq(i/(N*Ssize), i = 1 .. N)]:  
[>
```

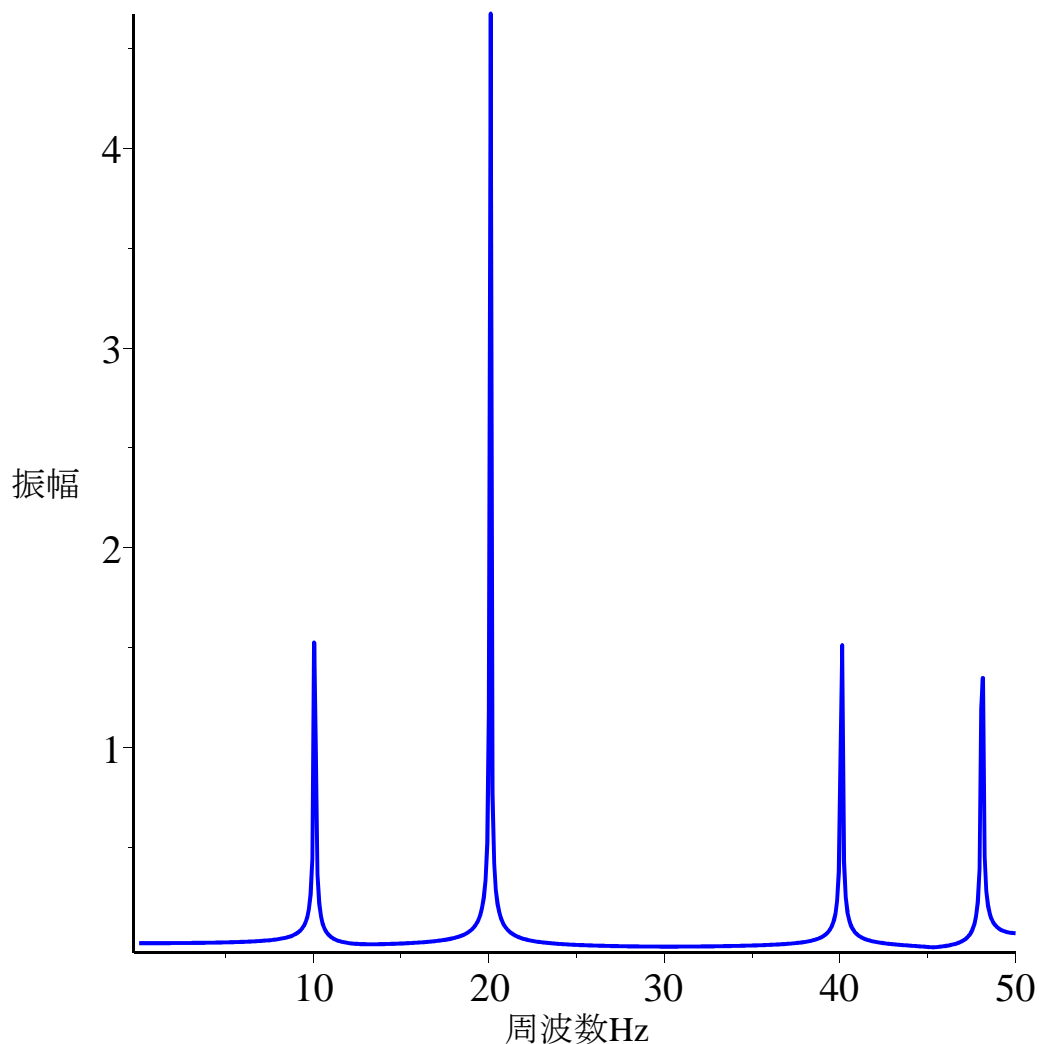
## ▼ フーリエ変換後のデータの可視化

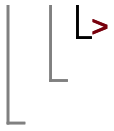
pointplot コマンドで FFT 変換により得られた周波数解析結果を描画します。

フーリエ変換結果の後半は前半と対称であり、前半のデータしか意味を持ちません。

入力の波とフーリエ変換した振幅を対応させるためには、算出した振幅を(サンプリング数/2)で割る必要があります。

```
> plots:-pointplot([seq([fqX[i], abs(fq)[i]/((1/2)*N)], i = 1 ..  
floor((1/2)*N)]), connect = true, numpoints = 2000, color = blue,  
labels = ["周波数Hz", "振幅"]);
```





無断転載禁止

Copyright © 2016 CYBERNET SYSTEMS CO., LTD. All rights reserved.