

심화수학 I 「학기일권」 탐구보고서

1학년 (1)반 (19)번 이름 : (정우용)

제목 : 뇌파 분석에서 세 가지 푸리에 변환 방식의 차이 분석

1. 탐구 동기 및 목적

본 탐구는 “수학으로 배우는 파동의 법칙” 책을 읽은 후 파동과 관련된 수학적 도구인 푸리에 변환에 대한 깊은 이해를 목표로 한다. 일상생활에서 흔히 접할 수 있는 파동의 원리를 수학적으로 분석하고, 이를 **실제 응용 분야에 접목시킬 수 있는 가능성**을 알아보기 위해서다. 특히, BCI 뇌파 분석과 같은 분야에서 푸리에 변환이 어떻게 활용되고 있는지 알아보려고 한다.

책을 읽으며 기본적인 푸리에 변환뿐 아니라 이산 푸리에 변환, 고속 푸리에 변환, 단시간 푸리에 변환 등 다양한 공식이 존재한다는 것을 알게 되었다. 여기서 어째서 이렇게 다양한 공식들이 존재하고, 각각의 변환 방식이 어떤 상황에서 어떤 장단점이 있는지 심도있게 탐구하고 싶었다. 이를 통해 뇌파 분석에 있어 **가장 적절한 변환 방식**을 선택하고 적용함으로써, 보다 정확하고 효율적인 분석 결과를 도출할 수 있을 것으로 기대한다.

따라서 본 탐구는 푸리에 변환의 다양한 형태를 비교 분석하고, 각각의 방식이 지닌 **고유의 특성과 이로 인한 장단점**을 명확히 하며, **실제 데이터 분석에 적용**해 보는 과정을 통해 이론적 지식과 실제 적용 사이의 간극을 좁히고자 한다.

2. 이론적 배경

1. 푸리에 급수

1.1 푸리에의 발견

같은 형태를 반복하는 주기를 가진 파동은, 아무리 복잡한 것이라도 단순한 파동이 여러 개 결합해 이루어진다. 이렇게 복잡한 파동을 단순한 파동으로 분해하여 생각하면 그 파동을 보다 간단하게 이해할 수 있다. 이때 단순한 파동은 삼각함수로 나타낸다.

1.2 삼각함수

1.2.1 sin 파동

원점이 중심이고 반지름의 길이가 1인 단위원이 있을 때, 그 위의 점(x, y)에 대하여 시초선에서 반시계 방향으로 회전을 각의 양의 방향으로 잡고, 그 각의 크기를 θ 라고 하면 $\sin\theta = y, \cos\theta = x, \tan\theta = \frac{y}{x}$

θ 가 변화함에 따라 점 P의 y좌표가 어떻게 변화하는지를 그래프로 나타낸 것이 sin 파동이다. 이때, 대부분의 파형은 수평축이 시간이므로, 각 그래프에 시간의 변화를 적용하고, 파장의 진폭까지 고려하면 다음과 같은 파동식을 얻게 된다. $f(x) = a\sin\omega t$

여러 개의 sin 함수가 더해져 만들어진 복합 파동은 다음과 같이 표현할 수 있다.

$$f(t) = a_1\sin\omega_1t + a_2\sin\omega_2t + a_3\sin\omega_3t + \dots + a_n\sin\omega_nt$$

1.2.1 cos 파동

θ 가 변화함에 따라 점 P의 x좌표가 어떻게 변화하는지를 그래프로 나타낸 것이 cos인 파동이며, 동일한 방법으로 cos 함수가 더해져 만들어진 복합 파동은 다음과 같이 표현할 수 있다.

$$f(t) = a_1\cos\omega_1t + a_2\cos\omega_2t + a_3\cos\omega_3t + \dots + a_n\cos\omega_nt$$

1.2.3 복합 파동

이때, sin 파동과 cos 파동은 0을 중심으로 진동하는 파동이므로, 평행이동한 파동을 나타내기 위해 상수

a_0 를 추가하면 복합 파동을 다음과 같이 나타낼 수 있다.

$$\begin{aligned} f(t) &= a_0 + b_1 \sin \omega_1 t + a_1 \cos \omega_1 t + b_2 \sin \omega_2 t + a_2 \cos \omega_2 t + \cdots + b_n \sin \omega_n t + a_n \cos \omega_n t \\ &= a_0 + \sum_{n=1}^{\infty} (a_n \cos n\omega t + b_n \sin n\omega t) \\ &= a_0 + \sum_{n=1}^{\infty} (a_n \cos \frac{2\pi n}{T} t + b_n \sin \frac{2\pi n}{T} t) \end{aligned}$$

이렇게 복잡한 파동을 단순한 파동으로 분해해서 공식으로 나타내고, 단순한 파동들을 더하여 복잡한 파동을 공식으로 표현할 수 있게 되었다. 여기서 구한 이 공식을 **푸리에 급수 공식**이라고 한다.

2. 푸리에 계수

2.1 함수의 직교성

삼각함수는 직교성을 갖는 함수이다. 이때 **함수의 직교성**은 **내적**을 이용해 알 수 있으며, 이는 다음과 같이 정의된다. $(f, g) = \int_a^b f(x)g(x)dx$ 즉, 곱해서 적분하면 된다. 이때 $\int_a^b f(x)$ 은 적분 기호로, 구간 $[a, b]$ 에서 $y = f(x)$ 그래프의 면적을 구하는 데 사용된다. 함수의 내적 값 역시 0이 된다면, 구간 $[a, b]$ 에서 $f(x), g(x)$ 는 직교한다. 이때, 면적은 +, -의 부호를 고려하기에 **내적이 0이 아닌 경우는 자기 자신을 내적하는 경우 뿐이다**. 이를 이용하면 a_n, b_n 을 구할 수 있다.

a_n, b_n , 즉 푸리에 계수는 함수의 직교성을 이용하여 다음과 같이 구할 수 있다.

$$a_0 = \frac{1}{T} \int_0^T f(t)dt \quad a_n = \frac{2}{T} \int_0^T f(t) \cos n\omega t dt \quad b_n = \frac{2}{T} \int_0^T f(t) \sin n\omega t dt$$

3. 오일러 공식

3.1 가우스 평면

가우스 평면이란 수직축에 허수, 수평축에 실수를 가진 평면으로, 함수를 이용해 가우스 평면 상에 그래프를 그릴 수 있다. $C(x) = a(x) + b(x)i$ 라고 했을 때, **가우스 평면 상에 반지름이 1인 원을 그리는 함수는 다음과 같다. $C(x) = \cos x + i \sin x$**

3.2 매클로린 전개

푸리에 급수와 같은 무한급수를 전개하고 싶지만 계수를 모를 때, 그 자체의 식에서 계수를 구해주는 것이 **매클로린 전개** 공식이다. 어떤 함수든지 다음과 같이 나타낼 수 있다. $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots$ 이때 $a_0 = f(0)$ 이며, 다항함수의 미분법을 고려하였을 때 **a_n 의 경우 $f(x)$ 를 n 차 미분해서 그 도함수에 0을 대입한 뒤, 양변을 $n!$ 으로 나눠서 구할 수 있다.**

$$a_n = \frac{1}{n!} f^{(n)}(0) \quad f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) x^n$$

이 공식을 이용하여 $f(x) = \sin x$ 와 $f(x) = \cos x$ 를 매클로린 전개하면 다음과 같은 식을 얻을 수 있다.

$$\sin x = 0 + x + 0x^2 - \frac{1}{3!}x^3 + 0x^4 + \frac{1}{5!}x^5 + 0x^6 + \cdots$$

$$\cos x = 1 + 0x - \frac{1}{2!}x^2 + 0x^3 + \frac{1}{4!}x^4 + 0x^5 + \cdots$$

즉, 이를 통해 $f(x) = \cos x + i \sin x$ 에서 $f^{(n)}(0)$ 의 값이 $1, i, -1, -i$ 패턴을 반복함을 알 수 있다. 이는 $f(x) = e^{ix}$ 과 동일하다. 따라서 $e^{ix} = \cos x + i \sin x$ 이며, 이것이 **오일러 공식**이다.

$$\begin{aligned} e^{i\omega t} &= \cos n\omega t + i \sin n\omega t, \quad e^{-i\omega t} = \cos n\omega t - i \sin n\omega t \\ \cos n\omega t &= \frac{1}{2}(e^{i\omega t} + e^{-i\omega t}), \quad \sin n\omega t = \frac{1}{2i}(e^{i\omega t} - e^{-i\omega t}) \end{aligned}$$

4. 푸리에 급수 전개의 복소 표현

오일러 공식을 푸리에 급수 및 계수에 대입 후 전개, 정리의 과정을 거치면 다음의 식을 얻을 수 있다. ($f(t)$: 연속 시간 신호, C_n : 푸리에 급수에서 각 주파수 성분 n 의 복소수 계수, $F(\omega)$: $f(t)$ 의 연속 시간 푸리

에 변환)

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{i\omega n t}, \quad C_n = \frac{1}{T} \int_0^T f(t) e^{-i\omega n t} dt$$
$$\therefore F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

3. 탐구 내용 및 과정

1. 푸리에 변환의 비교 분석

1.1 이산 푸리에 변환 (DFT)

1.1.1 DFT의 정의와 수학적 표현

이산 푸리에 변환(DFT)은 주어진 이산 시간 신호를 주파수 영역으로 변환하는 수학적 과정이다. 주어진 신호가 이산적이고 주기적인 경우, 그 신호는 복소수 지수 함수들의 합으로 표현될 수 있다. DFT는 이러한 표현을 가능하게 하는 변환이며, 신호의 기본 주파수 성분을 분석할 때 사용된다.

이산 시간 신호 $f(n)$ 에 대한 DFT는 다음과 같이 정의된다. $F(k) = \sum_{n=0}^N f(n) e^{-\frac{i2\pi}{N} kn}$

여기서 $f(n)$ 은 시간 영역에서 원래의 신호로, n 번째 시간 샘플 값을 의미하고, N 은 샘플의 총 개수로, 신호가 얼마나 많은 데이터 포인트를 포함하고 있는지를 의미한다. n 과 k 는 각각 이산 시간 샘플의 인덱스,

주파수 인덱스로, 0부터 $N-1$ 까지의 값을 가진다. 또한, 지수 $e^{-\frac{i2\pi}{N} kn}$ 는 n 번째 시간 샘플과 k 번째 주파수 성분 간의 위상 관계를 나타낸다.

1.1.2 DFT의 장단점 및 계산 효율성

DFT의 가장 큰 장점은 시간 영역에 존재하는 이산 신호의 주파수 구성을 분석할 수 있다는 것이다. 이를 통해 신호 처리 분야에서 필수적인 주파수 스펙트럼을 얻을 수 있으며, 다양한 응용 분야에서 신호의 특성을 이해하는 데 사용된다. 예를 들어, 소리의 피치를 분석하거나, 이미지에서의 패턴을 인식하는 데 있어 DFT는 필수적인 도구로 활용된다.

그러나 DFT를 직접 계산하는 과정에는 몇 가지 단점이 존재한다. 계산 복잡도가 $O(N^2)$ 으로, 신호의 샘플 수가 많을수록 필요한 연산의 수가 급격히 증가한다는 점이다. 이는 특히 대규모 데이터 세트를 다룰 때 계산 시간이 크게 증가하는 원인이 된다. 또한, 각 주파수 성분에 대한 계산이 독립적이지 않고 상호 연관되어 있어 병렬 처리에 어려움이 있다는 점도 단점으로 꼽힌다.

1.2 고속 푸리에 변환 (FFT)

1.2.1 FFT의 기본 원리와 알고리즘

고속 푸리에 변환(FFT)은 이산 푸리에 변환(DFT)의 계산을 효율적으로 수행하는 알고리즘이다. FFT는 신호의 주파수 성분을 분석하는 과정을 가속화하여 대규모 데이터 세트에 대한 DFT를 실시간으로 계산할 수 있게 한다. 이 알고리즘은 DFT의 계산 복잡도를 $O(N^2)$ 에서 $O(N \log N)$ 로 줄여 계산 시간을 단축한다.

FFT의 기본 원리는 시간 도메인의 이산 신호를 주파수 도메인으로 변환하는 DFT 공식과 동일하나, 계산 과정에서 중복 계산을 최소화한다. 주어진 신호 $f(n)$ 에 대해 FFT를 적용하면, 다음과 같은 수학적 형태로

표현할 수 있다. $F(k) = \sum_{n=0}^{N-1} f(n) W_N^{nk}$ 여기서 $W_N^{nk} = e^{-\frac{i2\pi}{N} nk}$ 는 N 차원 원시 단위근의 nk 제곱으로, DFT에서 복소 지수 함수를 대체한다. 이것이 FFT의 핵심적인 요소 중 하나이다.

1.2.2 FFT의 장단점 및 계산 효율성

FFT의 주된 장점은 DFT를 수행하는 데 필요한 연산량을 대폭 감소시킨다는 것이다. 이로 인해, 오디오 신호 처리, 이미지 처리, 통신 시스템에서 실시간 신호 분석이 가능해졌다. 특히, 긴 신호나 고해상도의 데이터에 대해 DFT를 계산할 때, FFT는 필수적인 도구로 자리잡았다.

FFT의 이론적 중요성은 **버터플라이 연산**이라고 불리는 기본 단계를 반복함으로써 신호의 주파수 성분을 분리하고, 각 성분을 **재귀적으로 결합**하는 방식에 있다. 이 과정은 DFT의 계산 과정을 **분할 정복 알고리즘**으로 전환하여, 각 단계에서 계산을 공유함으로써 전체적인 연산량을 감소시킨다.

그러나 FFT는 일정한 개수의 샘플을 가진 신호에 대해서만 최적으로 작동하며, 신호의 길이가 2의 거듭제곱이 아닐 경우 효율이 떨어질 수 있다. 또한, 실수와 복소수 데이터 모두에 적용 가능하지만, 복소수 데이터에 대해서는 계산량이 더 많아진다.

1.3 단시간 푸리에 변환 (STFT)

1.3.1 STFT의 정의와 수학적 표현

단시간 푸리에 변환(STFT)은 특히 **비정상 신호**를 분석할 때 유용하며, 변환 과정에서 신호를 겹치지 않는 구간으로 나누어 각 구간에 대한 푸리에 변환을 적용한다.

이는 다음과 같이 표현된다. $STFTf(t)(\tau, \omega) = \int_{-\infty}^{\infty} f(t) \cdot \omega(t - \tau) \cdot e^{-i\omega t} dt$

여기서 $f(t)$ 는 분석하고자 하는 시간 영역의 신호이고, $\omega(t - \tau)$ 는 신호에 적용되는 윈도우 함수이다.

1.3.2 STFT의 장단점 및 계산 효율성

STFT의 주된 장점은 신호의 시간에 따른 주파수 내용을 동시에 분석할 수 있다는 점이다. 이를 통해 신호가 시간에 따라 어떻게 변화하는지, 특정 시간 구간에서의 주파수 성분이 어떤지를 파악할 수 있다. 이는 음성 신호 처리, 음악 분석, 그리고 다른 다양한 신호 처리 응용 분야에서 매우 중요하다.

그러나 STFT는 신호를 얼마나 세밀하게 시간적으로 볼 수 있는지와 주파수적으로 얼마나 세밀하게 볼 수 있는지 사이에 **상충 관계**가 있다. 즉, 윈도우의 크기가 고정되어 있어 특정 주파수 해상도에 대한 시간 해상도가 제한되며, 이는 분석하고자 하는 신호의 세부적인 특성을 파악하는 데 제약을 줄 수 있다. 또한, STFT는 윈도우를 이동시키면서 많은 양의 중복 계산을 수행해야 하므로 계산량이 많아질 수 있다. 이러한 계산상의 부담은 특히 실시간으로 신호를 분석해야 하는 경우에 문제가 될 수 있다.

3. 데이터 분석 적용

3.1 실제 뇌파 데이터 준비

3.1.1 데이터 전처리 및 정제

<http://dx.doi.org/10.5524/100542>의 데이터 베이스에서 36번째 참가자의 뇌파 데이터를 활용했다. 데이터를 불러온 뒤, Common Average Referencing (CAR)을 사용해 잡음을 제거한다.

```
clear all; close all; clc;
addpath(genpath('D:\W00YONG\MATLAB\PhDYeom\Ground'));

% 파라미터 설정
sf = 1000; % 샘플링 주파수
ch_n = 62; % 채널 수
wnd_size = [-1 4]; % 윈도우 크기
baseline = [-1 0]; % 기준선
f_scale = 1; % 시간-주파수 분석을 위한 주파수 간격
freq_band = [0.1 100]; % 시간-주파수 분석을 위한 주파수 대역
normal = 1; % 기준선에 의한 정규화
fullscreen = get(0, 'ScreenSize'); % 모니터 화면 크기
[position] = EEG_62ch_layout_Brain_Products; % EEG 채널 위치

% 데이터 로드
load('sess01_subj36_EEG_MI.mat'); % 학습 데이터 로드
EEG = EEG_MI_train.x'; % EEG 데이터 단축 이름

% Common Average Referencing (CAR)을 사용한 재참조
EEG = EEG - repmat(mean(EEG,1), ch_n, 1);

% 이벤트 추출
events{1} = EEG_MI_train.t(find(EEG_MI_train.y_dec == 1)); % 오른손
events{2} = EEG_MI_train.t(find(EEG_MI_train.y_dec == 2)); % 왼손

% 데이터 에포킹
for i = 1:length(events)
    for tr = 1:size(events{i}, 2)
        e_EEG{i}(:, :, tr) = EEG(:, round(events{i}(tr) + (wnd_size(1) * sf)): round(events{i}(tr) + (wnd_size(2) * sf)));
    end
end
```

3.2 각 푸리에 변환 방식의 구현 및 실행

3.2.1 각 푸리에 변환 구현 및 분석 적용

다음과 같이 코드를 작성하였다. FFT는 내장 함수를 사용하고, DFT와 STFT는 직접 구현하였다.

```
% 실행 시간을 저장할 변수 초기화
FFT_times = [];
DFT_times = [];
STFT_times = [];

% FFT, DFT, STFT 구현
for i = 1:length(events)
    for tr = 1:size(e_EEG{i}, 3)
        % FFT
        tic;
        FFT_results{i}(:, :, tr) = fft(e_EEG{i}(:, :, tr), [], 2);
        FFT_times = [FFT_times, toc]; % 시간 저장

        % DFT
        tic;
        N = size(e_EEG{i}, 2);
        dft_matrix = exp(-2i * pi * (0:N-1).' * (0:N-1) / N);
        DFT_results{i}(:, :, tr) = e_EEG{i}(:, :, tr) * dft_matrix';
        DFT_times = [DFT_times, toc]; % 시간 저장

        % STFT
        tic;
        window_length = 256;
        for ch = 1:ch_n
            [S, F, T] = spectrogram(e_EEG{i}(ch, :, tr), window_length, window_length/2, window_length, sf);
            STFT_results{i}(ch, :, :, tr) = S;
        end
        STFT_times = [STFT_times, toc]; % 시간 저장
    end
end

% 결과를 표시할 채널 선택
channel_to_display = 1;

% FFT, DFT 및 STFT 결과 시각화 및 분석
for i = 1:length(events)
    figure('Position', fullscreen);
    for tr = 1:size(FFT_results{i}, 3)
        subplot(5, 10, tr);
        plot(abs(FFT_results{i}(channel_to_display, :, tr)));
        title([num2str(tr)]);
    end

    figure('Position', fullscreen);
    for tr = 1:size(DFT_results{i}, 3)
        subplot(5, 10, tr);
        plot(abs(DFT_results{i}(channel_to_display, :, tr)));
        title([num2str(tr)]);
    end

    figure('Position', fullscreen);
    for tr = 1:size(STFT_results{i}, 4)
        subplot(5, 10, tr);
        data_to_plot = abs(squeeze(STFT_results{i}(channel_to_display, :, :, tr)));
        imagesc(T, F, abs(STFT_results{i}(channel_to_display, :, :, tr)));
        title([num2str(tr)]);
        axis tight;
        colorbar;
    end
end
```

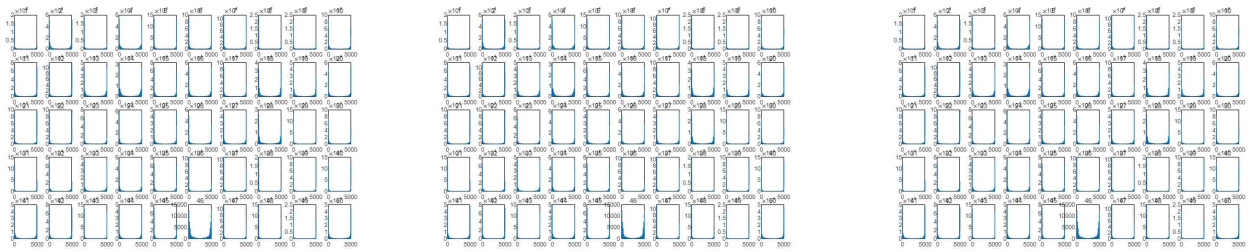
4. 탐구 결과 및 고찰

1. 결과 해석 및 보완할 점

1.1 결과 해석

그래프를 확인한 결과, 세 가지 모두 유사하게 출력되었음을 확인했다. 시간 효율성의 측면에서는 FFT가 시간 효율성이 가장 좋은 것으로 나타났다.

평균 FFT 실행 시간: 0.022894 초
평균 DFT 실행 시간: 0.790244 초
평균 STFT 실행 시간: 0.098286 초



푸리에 변환과 그 변형들이 제공한 결과의 유사성은 뇌파 데이터의 근본적인 특성을 반영한다. 뇌파는 연속적인 신호이며, 이산 푸리에 변환(DFT)과 고속 푸리에 변환(FFT)은 이러한 연속적인 신호를 샘플링하여 이산적인 주파수 영역으로 변환한다. 결과의 유사성은 해당 뇌파 데이터에 일관된 주파수 구성이 존재함을 나타낼 수 있지만, 세 가지 방식의 장단점 및 특징을 비교하고자 하는 목적을 달성시키기에는 어려움이 있다.

1.2 보완할 점

세 가지 방식 모두 유사한 결과가 나와, 제대로 장단점 및 특징을 분석할 수 없었다. 이를 해결하기 위해 다음과 같은 추가 연구가 필요할 것으로 보인다. 우선, STFT의 윈도우 크기와 중첩을 조정하여 시간 해상도를 개선할 수 있다. 또한, 웨이브릿 변환과 같은 다른 시간-주파수 분석 방법을 적용해 보는 것도 하나의 방법이다. 이를 통해 변환 방식이 신호를 어떻게 포착하는지 더 잘 이해할 수 있을 것이다.

그리고 결과 출력 및 계산 효율성 비교에서 그치지 않고, 시간-주파수 해상도 분석, 노이즈 감지 및 신호 분리 능력 평가도 진행된다면 더욱 유의미한 결과를 얻을 수 있을 것으로 보인다.

5. 제언 및 소감

'수학으로 배우는 파동의 법칙'을 읽고 나서 파동과 관련된 수학적 도구에 대한 이해를 깊게 할 수 있었다. 일상에서 자주 접하는 파동의 원리를 수학적으로 분석하고, 이를 실제 응용 분야, 특히 BCI 뇌파 분석에 접목시키는 과정이 흥미롭고, 즐거웠다. 비록 원하는 결과는 얻지 못하였지만, 이를 구현하기 위해 이론을 찾아보는 과정에서 푸리에 변환과 그 변형들인 이산 푸리에 변환, 고속 푸리에 변환, 단시간 푸리에 변환 등의 다양한 형태들이 왜 존재하는지, 그리고 각각이 특정 상황에서 어떤 장단점을 가지는지 심도 있게 탐구해볼 수 있었다는 것은 큰 의의가 있다고 생각한다. 또한, 실제 뇌파 데이터를 분석하여 시간 효율성을 직접 출력해 보는 과정이 유의미하다고 느낀다.

이론에서 실제 적용까지의 과정에서 가장 적절한 변환 방식을 선택하고 적용함으로써 뇌파 분석의 정확성과 효율성을 증대시킬 수 있을 것으로 시사된다. 이러한 과정을 통해, 이론적 지식과 실제 적용 사이의 간극을 좁히려는 노력이 의미가 있다고 생각한다. 이 탐구는 앞으로의 과학적 탐구에 대한 열정을 불러일으키고, 더 나아가 이 분야의 추가적인 연구와 발전에 대한 기대를 키우는 계기가 되었다. 기회가 된다면 추가 연구를 진행하여 세 가지 방식의 특징을 보다 명확히 이해하고자 한다.