РАБОТА №9. НЕПРЕРЫВНОЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЕ ДИСКРЕТНЫХ СИГНАЛОВ

Цель работы: изучение возможностей применения вейвлет-преобразования при обработке сигналов.

Планируемая продолжительность: от 2 до 4 академических часов.

Тип работы: с использованием компьютерных средств.

Теоретические основы

Непрерывное вейвлет преобразование (НВП) – еще один инструмент анализа сигнала, а также его сжатия. Для сигнала $f(t)$ НВП определяется следующим образом:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \cdot \psi\left(\frac{t-b}{a}\right) dt, \tag{13.1}$$

где $\psi\left(\frac{t-b}{a}\right)$ - вейвлет, используемый для анализа;

a – параметр масштаба вейвлета;

b – параметр сдвига вейвлета.

То есть, вычислив интеграл (13.1) для какого-либо сигнала получим двумерную поверхность с величинами a и b в качестве параметров. Принято обозначать по оси абсцисс параметр a, по оси ординат – b, а $W(a,b)$, который называется вейвлет-коэффициентом при данных a,b. Соответственно результат вычисления (13.1) – набор вейвлет-коэффициентов при всех допустимых a и b.

С увеличением a вейвлет сжимается вдоль оси абсцисс (и наоборот), с увеличением b – сдвигается вдоль сигнала. При сжатии вейвлета, в сущности, увеличивается частота присутствующих в нем составляющих (так как при сжатии уменьшается период), поэтому можно сказать, что **ось *a* на графике вейвлет-коэффициентов сопоставима с осью частот, а ось *b* – с осью времени.** То есть, НВП относится к классу частотно-временных преобразований, позволяющих анализировать изменение частотного состава сигнала с течением времени.

Можно обратить внимание, что при конкретных значениях a (в данном примере a=1) выражение (13.1) принимает знакомый вид:

$$W(1,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \cdot \psi(t-b)\, dt,$$

что соответствует ВКФ сигнала и вейвлета. Следовательно, НВП является множеством функций взаимной корреляции сигнала с вейвлетом при различных масштабах a последнего.

Для дискретных сигналов, за счет замены интеграла на сумму, (13.1) примет вид:

$$W(a,b) = \frac{\Delta t}{\sqrt{a}} \sum_{j=1}^{N} S_j \cdot \psi\left(\frac{j \cdot \Delta t - b}{a}\right), \quad (13.2)$$

где Δt - интервал дискретизации;

S - дискретный сигнал.

В качестве вейвлета может быть использован целый ряд функций, допускается даже использование собственной функции в качестве вейвлета. Чтобы быть вейвлетом, функция должна удовлетворять некоторым требованиям, которые здесь описываться не будут. На практике чаще всего используют МНАТ-вейвлет (Mexican Hat, мексиканская шляпа):

$$\psi(t) = (1 - t^2) e^{-\frac{t^2}{2}}. \quad (13.3)$$

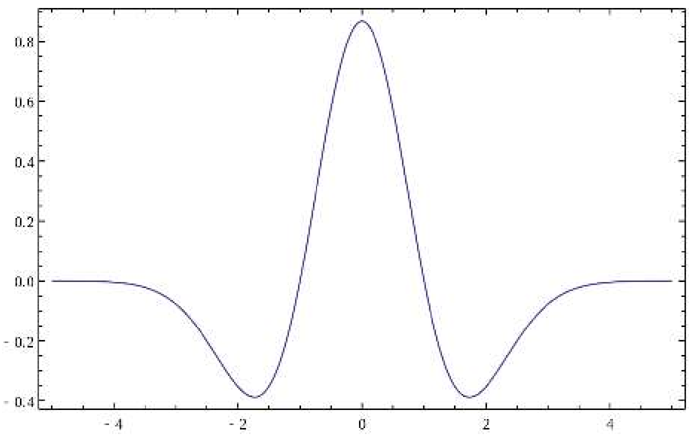

Рис. 9.1. МНАТ-вейвлет

Для визуализации НВП принято использовать так называемую скалограмму, значения для которой вычисляются на основе вейвлет-коэффициентов:

$$Sk(a,b) = |W(a,b)|^2. \quad (13.4)$$

Ниже представлен пример сигнала и его скалограммы.

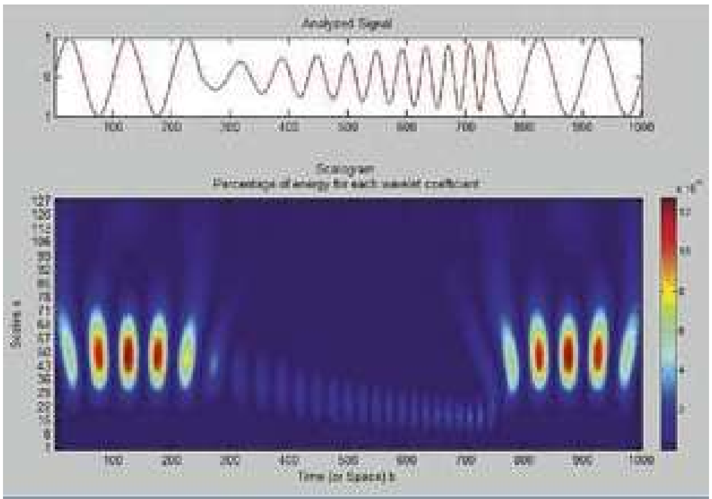

Рис. 9.2. Сигнал и его скалограмма

Непрерывное вейвлет-преобразование в Python

Как и до этого импортируем зависимости, переносим plot, и добавляем ещё одну функцию отрисовки графиков plot_scalogram, как следует из её названия при помощи неё будет рисовать график вейвлет коэффициентов.

```
import numpy as np
import matplotlib.pyplot as plt
from typing import Callable
from scipy.fft import fftshift, fft, fftfreq
%matplotlib widget
```

Рис. 9.3. Импорт зависимостей

```
def plot(*args, y=None, stem=False) -> None:
    ax = plt.figure()
    if y is None:
        for y in args:
            x = np.arange(y.size)
            if stem:
                plt.stem(x, y)
            else:
                plt.plot(x, y)
    else:
        if stem:
            plt.stem(args[0], y)
        else:
            plt.plot(args[0], y)
    plt.grid(True)
    plt.ylabel('y')
    plt.xlabel('x')
    plt.show()
```

Рис. 9.4. Функция построения графиков

```python
def plot_scalogram(wavelet_coeficients: np.ndarray) -> None:
    xlist = ylist = np.arange(0, wavelet_coeficients.shape[0])
    X, Y = np.meshgrid(xlist, ylist)
    Z = wavelet_coeficients
    fig, ax = plt.subplots(1, 1)
    cp = ax.contourf(X, Y, Z)
    fig.colorbar(cp, label='Значения вейвлет-коэффициентов')
    ax.set_title('Скалограмма сигнала')
    ax.set_ylabel('Ряды')
    ax.set_xlabel('Отсчёты')
    plt.show()
```

Рис. 9.5. Функция построения графика скалограммы

Для реализации НВП необходимо сперва задать вейвлет. В данной работе ограничимся MHAT-вейвлетом.

```python
def mexican_hat(t: np.ndarray):
    return np.exp((-1) * np.power(t, 2) / 2) * (1 - t ** 2)
```

```python
t = np.linspace(start=-6, stop=6)
plot(mexican_hat(t))
```

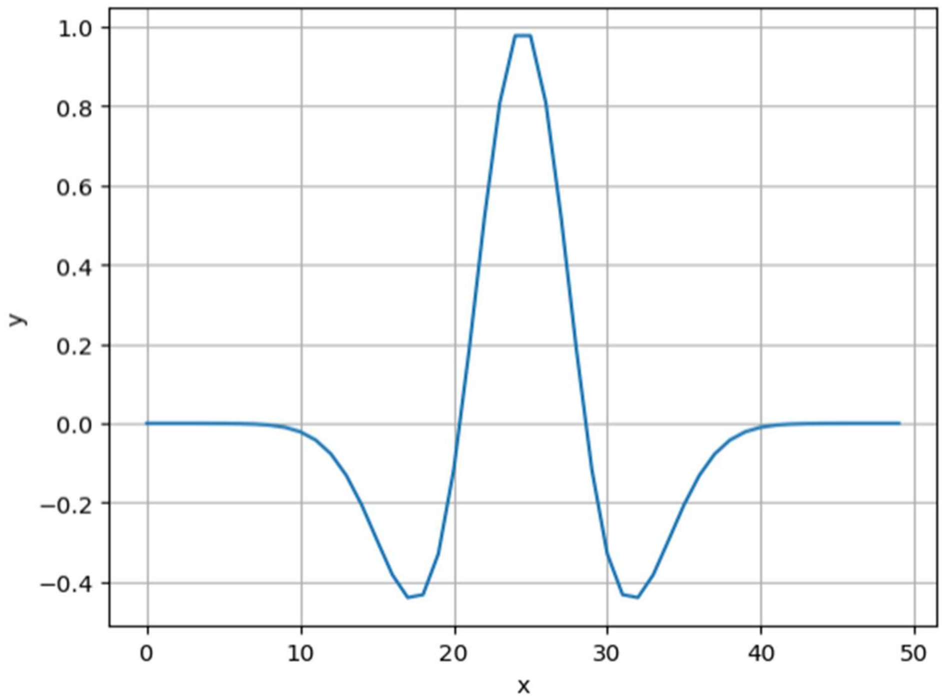

Рис. 9.6. MHAT-вейвлет в Smath

Зададим анализируемый сигнал. Чтобы увидеть основное отличие ДПФ от НВП, в сигнале предусмотрено резкое повышение частоты.

```
def y(x: np.ndarray) -> np.ndarray:
    result = np.zeros(x.size)
    for idx in range(result.size):
        if x[idx] < np.pi:
            result[idx] = np.sin(0.5 * 2 * np.pi * x[idx])
        else:
            result[idx] = np.sin(0.8 * 2 * np.pi * x[idx])
    return result
```

```
N = 30
x = np.arange(start=1, stop=30)
signal = y(x * (2 * np.pi / N))
plot(signal)
```

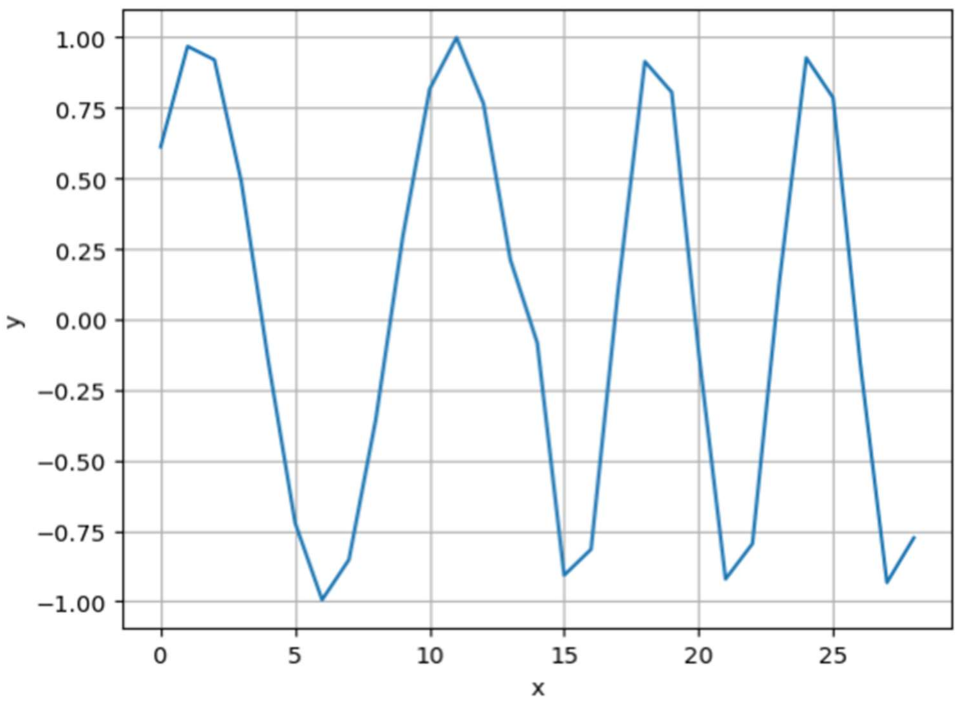

Рис. 9.7. Анализируемый сигнал

Теперь можно сформировать выражение для самого НВП. Как было подмечено выше, НВП при конкретных значениях масштаба является ВКФ вейвлета и сигнала. Воспользуемся этими данными и на основе спроектированного в предыдущих работах выражения для ВКФ зададим в Smath выражение (13.2), как это представлено на рисунке ниже.

```python
def wavelet_transform(signal: np.ndarray,
                      psi_func: Callable):
    dt = 2 * np.pi / signal.size

    def wavelet_transform_array(a: float):

        def wavelet_transform_element(b: float):
            result_sum = np.zeros(signal.size)
            for idx in range(result_sum.size):
                result_sum[idx] = signal[idx] * psi_func(((idx + 1) * dt - b) / a)
            return np.sum(result_sum)

        result_array = np.zeros(signal.size)
        for idx in range(result_array.size):
            result_array[idx] = wavelet_transform_element((idx + 1) * dt)
        return result_array

    result = []
    for k in range(signal.size):
        result.append(wavelet_transform_array((k + 1) * dt))
    return np.array(result)
```

Рис. 9.8. Задание НВП

Результат построения скалограммы для описанного выше примера на рисунке.

```python
plot_scalogram(wavelet_transform(signal, mexican_hat))
```

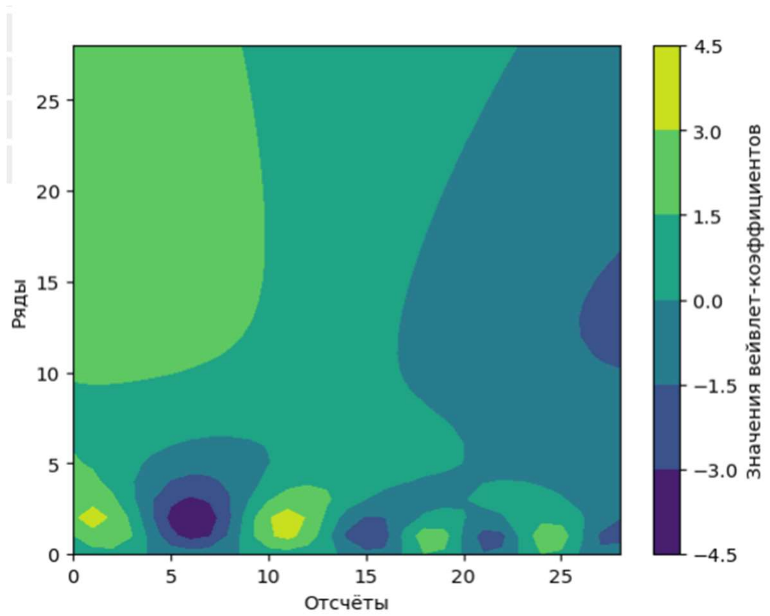

Рис. 9.9. Скалограмма сигнала

Из графика коэффициентов видно, что при малых значениях a, что соответствует большому масштабу вейвлета, мелких деталей выявлять не удается, но четко видна граница перехода сигнала из одной частоты в другую. Причем, при масштабе, которому соответствует приблизительно 9й ряд, эта граница по временному положению совпадает с ее положением в самом сигнале. Идентифицировать данный участок как границу двух частот легко – при масштабе большем граница смещается в сторону, а при меньшем начинают проявляться более мелкие детали, соответствующие отдельным полупериодам синусоиды (то есть в подобных случаях следует искать такой масштаб, при котором за счет совсем уж мелких деталей, таких как отдельные вершины, граница частот не уводится в сторону, и при котором детализация не столь мала, что граница «уплывает» или размывается). Подобный случай рассматривается на рисунке ниже (в сущности, совпадает с нашим премером, но здесь коэффициентов больше, сигнал большей длительности, за счет чего результат выглядит плавнее, присутствуют полутона).

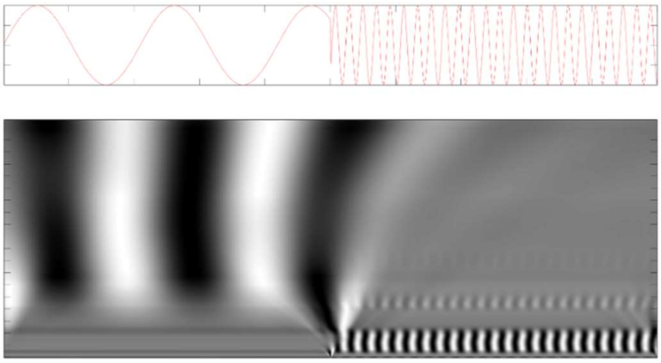

Рис. 9.10. Пример резкого перехода частот

Теперь повторно построим скалограмму, но только в этот раз каждое значения возьмём абсолютно, прибавим 50, так как значения вейвлет коэффициентов будут меньше нуля, следовательно при возведении в степень большая часть данных будет сглажена. И соответственно после суммирования возведём во 2 степень полученные данные

```
plot_scalogram(np.power(np.abs(wavelet_transform(signal, mexican_hat)) + 50, 2))
```

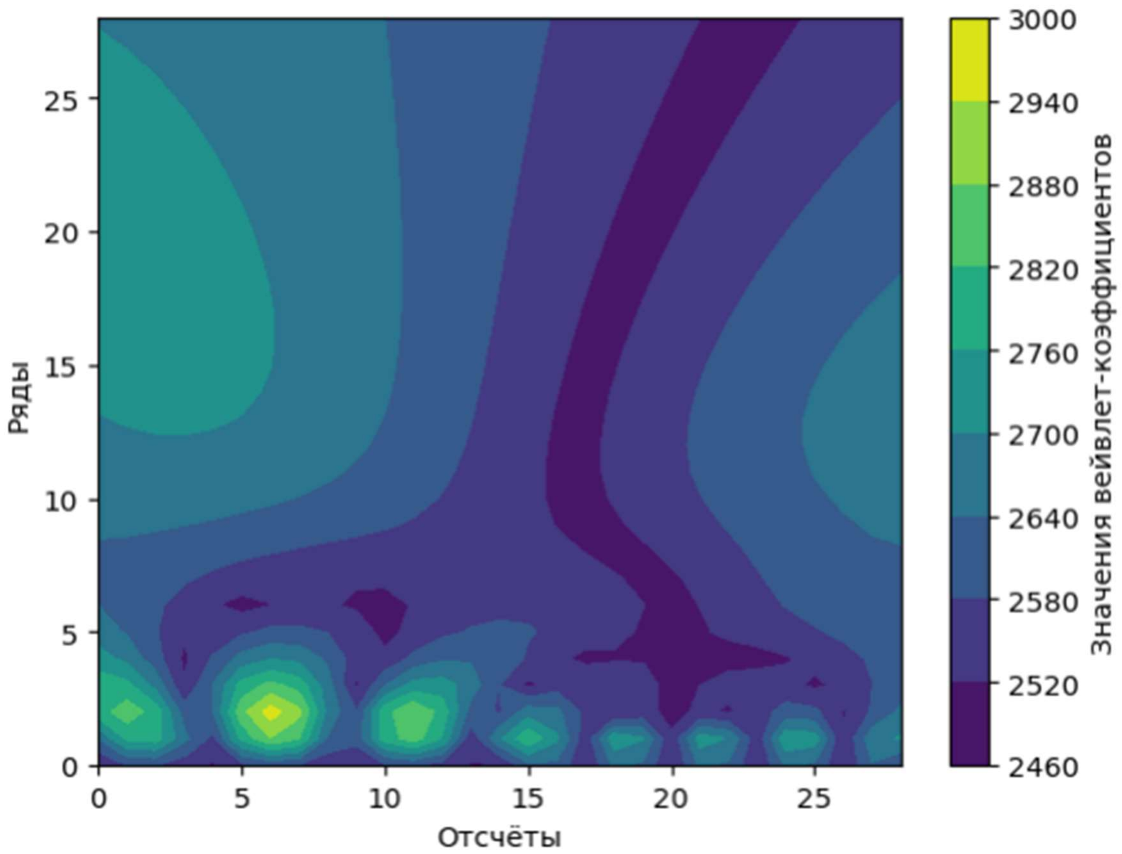

Рис. 9.11. Скалограмма

Граница частот видна еще контрастнее из-за модуля и квадратирования.

Сравним скалограмму с частотным спектром. Для этого зададим выражения для его реализации, рассмотренные в прошлых работах.

```
N = 30
T = (2 * np.pi / N)
x = np.linspace(0.0, N * T, N, endpoint=False)
yf = fft(y(x))
xf = fftfreq(N, T)[:N // 2]
```

Рис. 9.12. Выражения для реализации спектра

И построим соответствующий график.

```
plot(xf, y=2.0 / N * np.abs(yf[0:N // 2]))
```

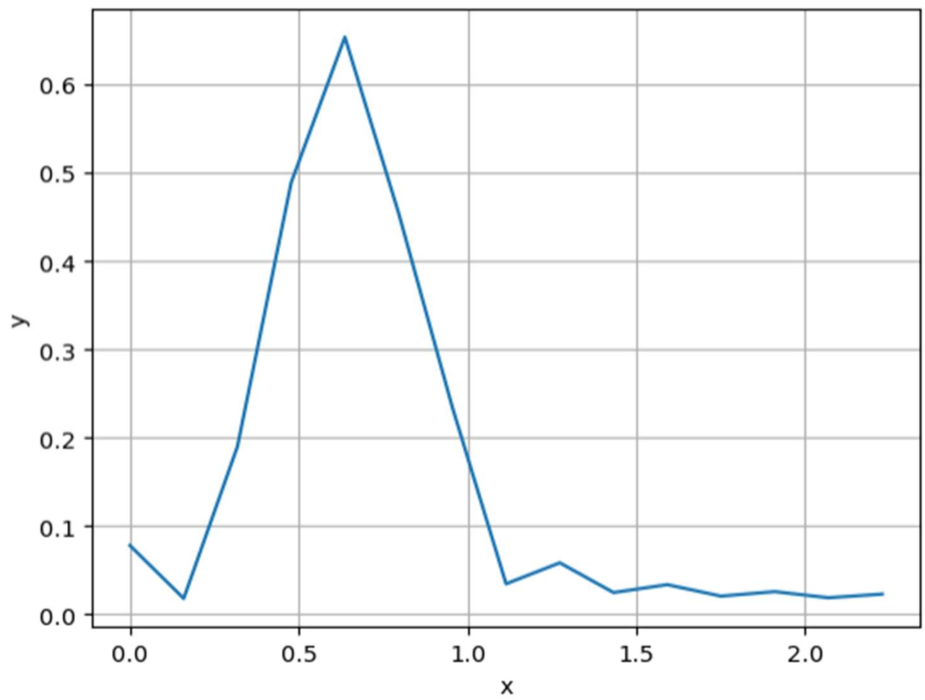

Рис. 9.13. График спектра

Можно видеть, что из-за близости частот на спектре они объединяются. При этом данный график не представляет никакой информации об изменении во времени спектра сигнала, в отличие от скалограммы (или самих вейвлет-коэффициентов).

Задание 1

1. Задать сигнал в соответствии с вариантом, построить его график
2. Задать MHAT-вейвлет, построить его график
3. Задать выражения для реализации НВП
4. Сформировать массив вейвлет-коэффициентов заданного сигнала
5. Построить в нем как график коэффициентов, так и скалограмму.
6. Построить спектр сигнала по ДПФ.

7. Записать вывод об отличиях и схожести получения спектра по ДПФ и скалограммы вейвлет-преобразования

Таблица 13.1.

Варианты заданий

Варианты	**Сигнал**
1,6,11, 16,21,26	$y(x) := \text{if } x < \pi$ $\sin(0{,}5 \cdot 2 \cdot \pi \cdot x) + \sin(0{,}8 \cdot 2 \cdot \pi \cdot x)$ else $\sin(0{,}8 \cdot 2 \cdot \pi \cdot x)$
2,7,12, 17,22,27	$y(x) := \text{if } x < \pi$ $\sin(0{,}5 \cdot 2 \cdot \pi \cdot x)$ else $\sin(0{,}8 \cdot 2 \cdot \pi \cdot x) + \sin(0{,}5 \cdot 2 \cdot \pi \cdot x)$
3,8,13, 18,23,28	$y(x) := \text{if } x < \pi$ $\cos(0{,}5 \cdot 2 \cdot \pi \cdot x)$ else $\cos(0{,}8 \cdot 2 \cdot \pi \cdot x)$
4,9,14, 19,24,29	$y(x) := \text{if } x < \pi$ $\cos(0{,}5 \cdot 2 \cdot \pi \cdot x) + \cos(0{,}8 \cdot 2 \cdot \pi \cdot x)$ else $\cos(0{,}8 \cdot 2 \cdot \pi \cdot x)$
5,10,15, 20,25,30	$y(x) := \text{if } x < \pi$ $\cos(0{,}5 \cdot 2 \cdot \pi \cdot x)$ else $\cos(0{,}8 \cdot 2 \cdot \pi \cdot x) + \cos(0{,}5 \cdot 2 \cdot \pi \cdot x)$