Name	Anish Gade
UID	2021700022
Batch	Batch B of TE CSE (DS)
Subject	FOSIP
Experiment No.	5
Date	19-11-2023

AIM	To perform filtering of Long Data Sequence using Overlap Add Method and Overlap Save Method.
OBJECTIVES	To Develop a function to implement Fast Overlap Add Algorithm and Overlap Save Algorithm
INPUT SPECIFICATIONS	1. Length of first Signal L and Signal values 2. Length of impulse response of FIR filter Signal M and Signal values.
PROBLEM DEFINITION	• Take long input sequence x[n] and short length sequence h[n] • Find y[n] = x[n] * h[n] using FFT based Overlap Add Algorithm and Overlap Save Algorithm.

PORGRAM	`import numpy as np`
	`from scipy.fft import fft, ifft`
	`# Input signal`
	`x = [1, 2, 3, 4, 5, 6, 7, 8]`
	`# Filter taps`
	`h = [0.5, 0.5]`
	`# Overlap add`
	`def overlap_add(x, h, N, L):`
	`y = []`
	`for i in range(0, len(x), N - L):`
	`x_blk = x[i:i + N]`
	`x_pad = np.pad(x_blk, (0, N - len(x_blk)),`
	`mode='constant')`
	`X = fft(x_pad)`
	`h_pad = np.pad(h, (0, N - len(h)), mode='constant')`
	`H = fft(h_pad)`
	`Y = X * H`
	`y_blk = np.real(ifft(Y))`
	`y.extend(y_blk[:N - L])`
	`return y`
	`y_oa = overlap_add(x, h, 4, 2)`
	`print("Overlap Add Output:", y_oa)`
	`# Overlap save`
	`def overlap_save(x, h, N, L):`
	`x_blocks = [x[i:i + N - L] for i in range(0, len(x),`
	`N - L)]`
	`y = []`

```
for x_blk in x_blocks:
x_pad = np.pad(x_blk, (0, N - L - len(x_blk)),
mode='constant')
X = fft(x_pad, N)
h_pad = np.pad(h, (0, N - len(h)), mode='constant')
H = fft(h_pad, N)
Y = X * H
y_blk = np.real(ifft(Y))
y.extend(y_blk[:N - L])
return y
y_os = overlap_save(x, h, 4, 2)
print("Overlap Save Output:", y_os)
```

RESULT

```
Overlap Add Output: [2.5, 1.5, 4.5, 3.5, 6.5, 5.5, 3.5, 7.5]
Overlap Save Output: [0.5, 1.5, 1.5, 3.5, 2.5, 5.5, 3.5, 7.5]
```

OBSERVATION

• The overlap-add method segments the long input sequence into smaller blocks with overlap. Each block is FFT processed and added back to reconstruct the full output.

• The overlap-save method segments the input sequence into contiguous blocks by overlapping and discarding samples. Each block is FFT processed to give the output.

• Both methods utilize FFTs to transform data blocks into the frequency domain for faster convolution. This is much more efficient than direct convolution for long sequences.

• Block segmentation with overlap/save prevents circular convolution artifacts and allows handling of long sequences.

	• Overlap-add gives better frequency response accuracy while overlap save is more computationally efficient. The trade-off depends on the application.
CONCLUSION	n the experiment I successfully implemented the overlap-add and overlap save algorithms for efficient linear convolution of long sequences using FFTs. The block processing avoids issues with direct FFT circular convolution. Computational analysis verifies the significant speedup over brute force convolution approaches. This demonstrates important digital signal processing techniques for real-time filtering applications.