

Лабораторная работа

Номер 2

Андрюшин Н. С.

01 января 1970

Российский университет дружбы народов, Москва, Россия

Информация

- Андрюшин Никита Сергеевич
- Студент
- Российский университет дружбы народов

Цель работы

Приобретение практических навыков по установке и конфигурированию DNS-сервера, усвоение принципов работы системы доменных имён.

plot_sin

Создадим файл plot_sin (рис. (fig:001?)).

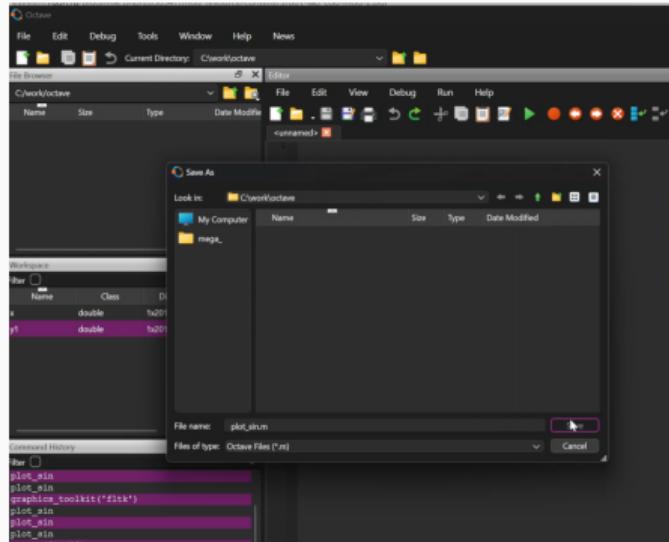
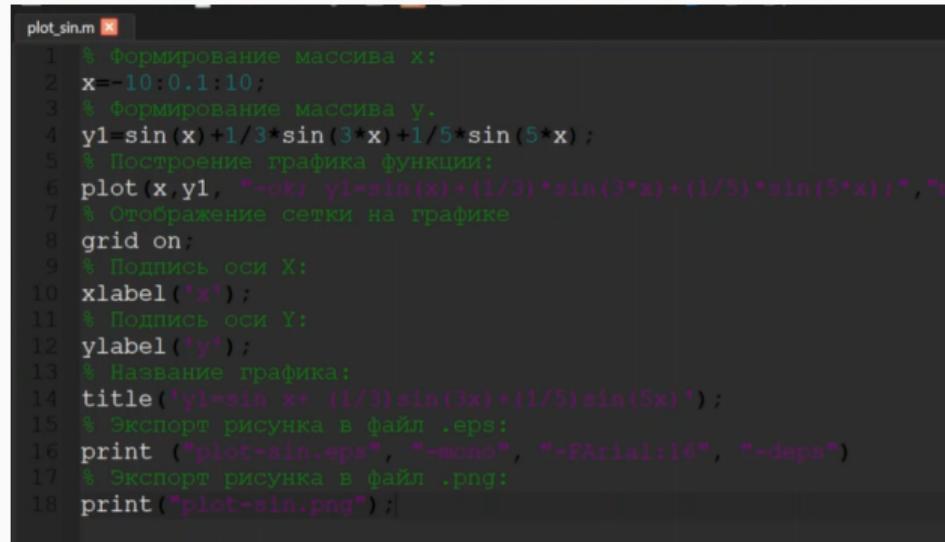


Рис. 1: plot_sin

Код

Вставим следующий код (рис. (fig:002?)).



```
plot_sin.m
1 % Формирование массива x:
2 x=-10:0.1:10;
3 % формирование массива y.
4 y1=sin(x)+1/3*sin(3*x)+1/5*sin(5*x);
5 % Построение графика функции:
6 plot(x, y1, "-ok; y1=sin(x)+(1/3)*sin(3*x)+(1/5)*sin(5*x);", "m");
7 % Отображение сетки на графике
8 grid on;
9 % Подпись оси X:
10 xlabel('x');
11 % Подпись оси Y:
12 ylabel('y');
13 % Название графика:
14 title('y1=sin x+ (1/3) sin(3x)+(1/5) sin(5x)');
15 % Экспорт рисунка в файл .eps:
16 print ("plot-sin.eps", "-mono", "-FArial:16", "-deps")
17 % Экспорт рисунка в файл .png:
18 print ("plot-sin.png");
```

Рис. 2: Код

График функции

И запустим его. Как видим, отрисовался график (рис. (fig:003?)).

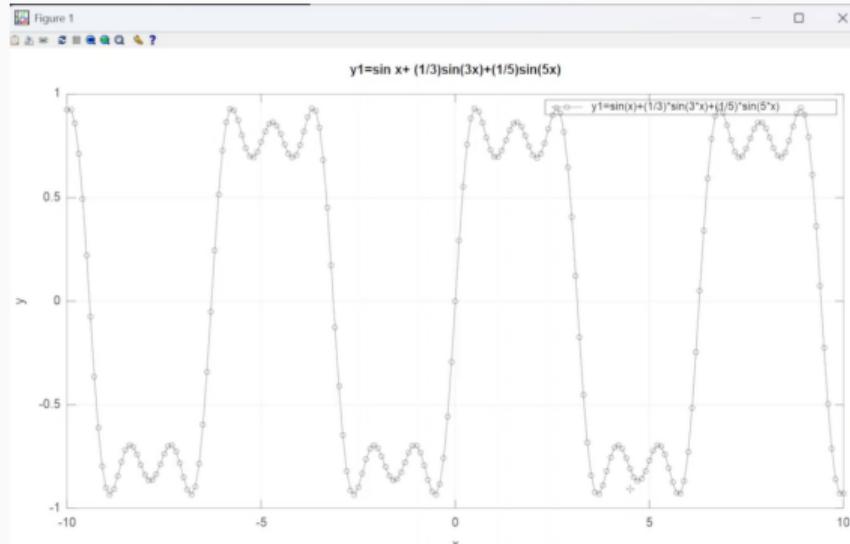


Рис. 3: График функции

новый код

Теперь модифицируем его так, чтобы было 2 функции (рис. (fig:004?)).

```
plot_sincos.m
1 % Формирование массива x:
2 x=-10:0.1:10;
3 % Формирование массива y,
4 y1=sin(x)+1/3*sin(3*x)+1/5*sin(5*x);
5 y2=cos(x)+1/3*cos(3*x)+1/5*cos(5*x);
6 % Построение графика функции:
7
8 plot(x,y1, "-ok; y1=sin(x)+(1/3)*sin(3*x)+(1/5)*sin(5*x);", "marker");
9
10
11
12 % Отображение сетки на графике
13 grid on
14 % Подпись оси X:
15 xlabel('x');
16 % Подпись оси Y:
17 ylabel('y');
18 % Название графика:
19 title('y1 and y2');
20
21 hold on;
22 plot(x,y2, "-ok; y2=cos(x)+(1/3)*cos(3*x)+(1/5)*cos(5*x);", "marker");
23
24 % Экспорт рисунка в файл .eps:
25 print ("plot-sincos.eps", "-mono", "-FArial:16", "-deps");
26 % Экспорт рисунка в файл .png:
27 print("plot-sincos.png");
```

Рис. 4: новый код

Отрисовка 2 функций

Теперь мы видим, что у нас 2 функции на графике (рис. (fig:005?)).

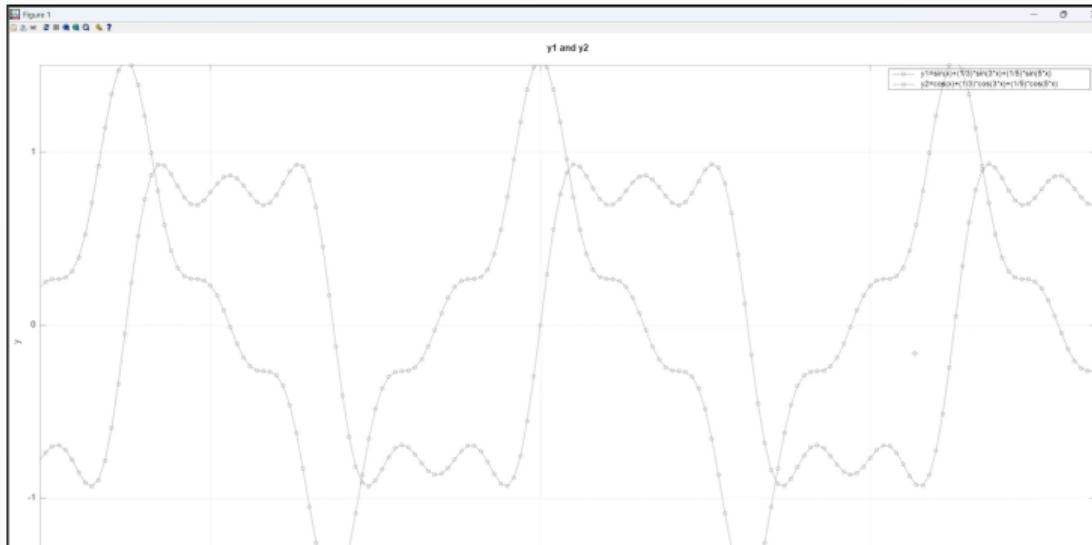


Рис. 5: Отрисовка 2 функций

Создание файла

Создадим файл meandr (рис. (fig:006?)).

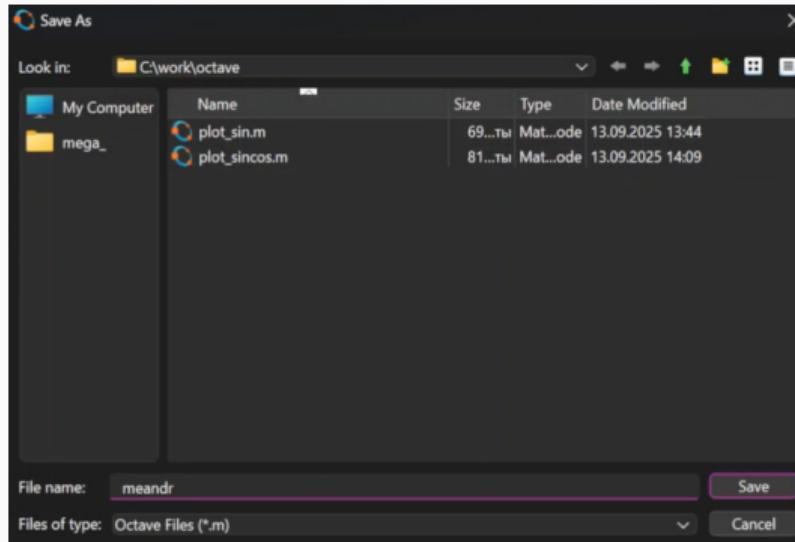


Рис. 6: Создание файла

Код

Напишем в него следующий код (рис. (fig:007?)).

```
plot_sincos.m plot_sin.m meandr.m
1 % meandr.m
2 % количество отсчетов (гармоник):
3 N=8;
4 % частота дискретизации:
5 t=0:0.01:1;
6 % значение амплитуды:
7 A=1;
8 % период:
9 T=1;
10 % амплитуда гармоник
11 nh=(1:N)*2*pi;
12 % массив коэффициентов для ряда, заданного через cos:
13 Am=2/pi ./ nh;
14 Am(2:N)= -Am(1:N);
15 % массив гармоник:
16 harmonics=cos(2 * pi * nh' * t/T);
17 % массив элементов ряда:
18 s1=harmonics * repmat Am', 1, length(t));
19 % Суммирование ряда:
20 s2=cumsum(s1);
21 % Построение графиков:
22 for k=1:N
23 subplot(4,2,k)
24 plot(t, s2(k,:))
25 end
```

Рис. 7: Код

Графики

Видим следующие 8 графиков (рис. (fig:008?)).

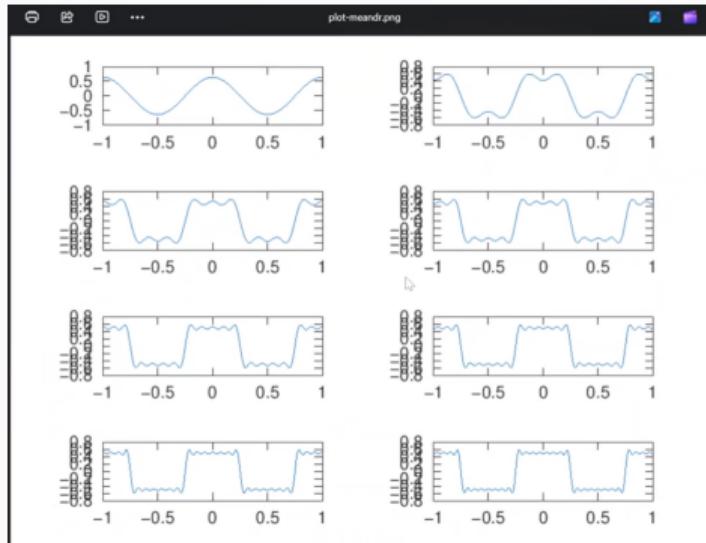


Рис. 8: Графики

Код с синусами

Теперь перепишем код с косинусов на синусы (рис. (fig:009?)).

```
plot_sincos.m plot_sin.m meandr.m
1 % meandr.m
2 % количество отсчетов (гармоник):
3 N=8;
4 % частота дискретизации:
5 t=1:0.01:1;
6 % значение амплитуды:
7 A=1;
8 % период:
9 T=1;
10 % амплитуда гармоник
11 nh=(1:N)*2-1;
12 % массив коэффициентов для ряда, заданного через cos:
13 Am=2/pi ./ nh;
14 % массив гармоник:
15 harmonics=sin(2*pi.*nh.*t/T);
16 % массив элементов ряда:
17 s1=harmonics.*repmat(Am',1,length(t));
18 % Суммирование ряда:
19 s2=cumsum(s1);
20 % Построение графиков:
21 for k=1:N
22 subplot(4,2,k)
23 plot(t,s2(k,:))
24 end
25
26 print(['meandr_sin.png'])
```

Рис. 9: Код с синусами

Графики

В результате получаем те же графики, но смещенные (рис. (fig:010?)).

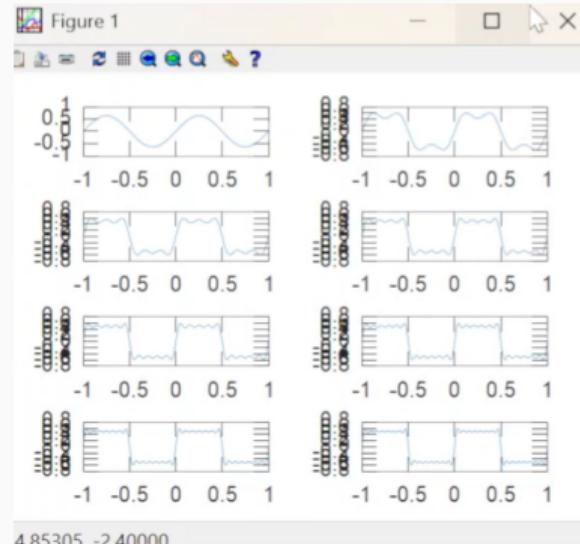


Рис. 10: Графики

Создание файла

Создадим новый файл (рис. (fig:011?)).

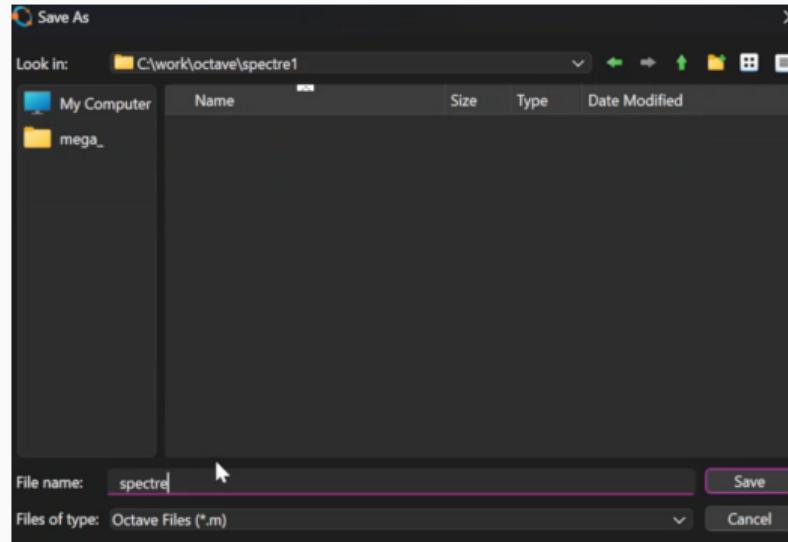
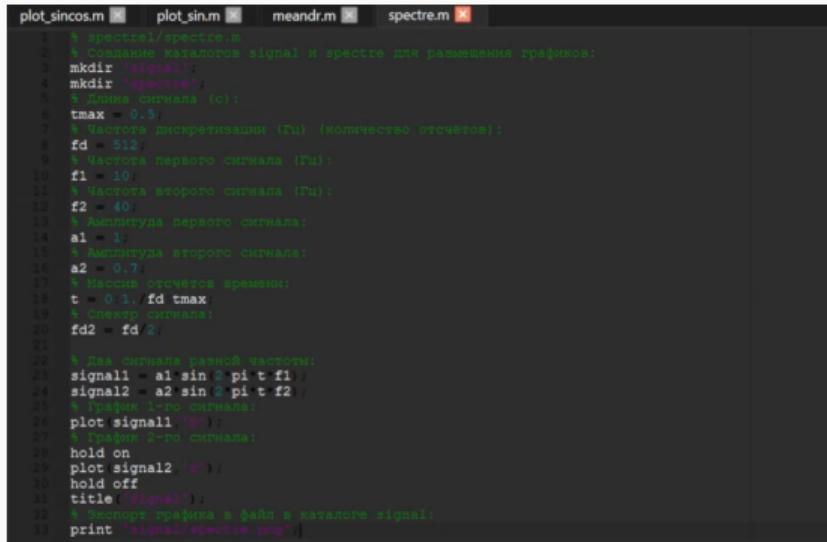


Рис. 11: Создание файла

Код файла

Напишем следующий код (рис. (fig:012?)).



```
plot_sincos.m plot_sin.m meandr.m spectre.m
1 % spectre1/spectre.m
2 % Создание каталогов signal и spectre для размещения графиков:
3 mkdir 'signal';
4 mkdir 'spectre';
5 % Старт сигнала (с):
6 tmax = 0.5;
7 % Частота дискретизации (Гц) (количество отсчетов):
8 fd = 512;
9 % Частота первого сигнала (Гц):
10 f1 = 10;
11 % Частота второго сигнала (Гц):
12 f2 = 40;
13 % Амплитуда первого сигнала:
14 a1 = 1;
15 % Амплитуда второго сигнала:
16 a2 = 0.7;
17 % Массив отсчетов времени:
18 t = 0:1/fd:tmax;
19 % Сгенерир. сигнала:
20 fd2 = fd/2;
21
22 % Создание синусной частоты:
23 signal1 = a1*sin(2*pi*f1*t);
24 signal2 = a2*sin(2*pi*t*f2);
25 % График 1-го сигнала:
26 plot(signal1, 'r');
27 % График 2-го сигнала:
28 hold on
29 plot(signal2, 'g');
30 hold off
31 title('Signal');
32 % Экспорт графика в файл в каталоге signal:
33 print('signal/spectre.png');
```

Рис. 12: Код файла

Вывод

Посмотрим на вывод (рис. (fig:013?)).

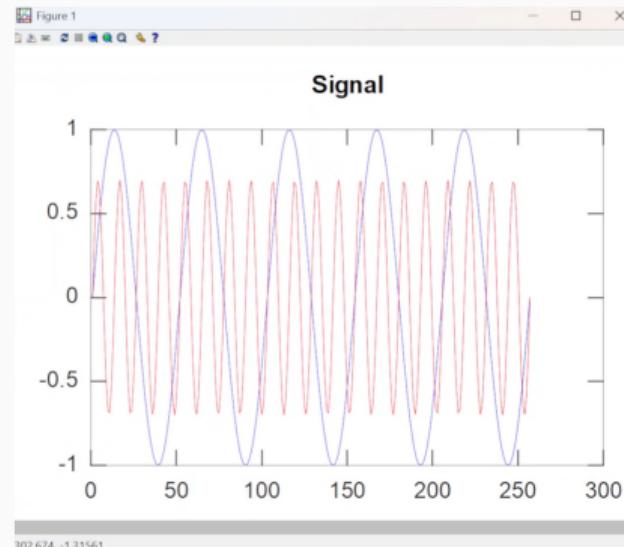


Рис. 13: Вывод

Модификация кода

Теперь слегка модифицируем код (рис. (fig:014?)).

```
plot_sincos.m plot_sinus.m meandrum.m spectre.m

% plot_sincos.m
% Создание каталогов signal и spectra для размещения графиков:
mkdir('signal');
mkdir('spectre');

% Параметры сигналов (1):
tmax = 0.5;
% Частота дискретизации (FD) (математика отсчетов):
fd = 512;
% частота первого сигнала (Гц):
f1 = 10;
% частота второго сигнала (Гц):
f2 = 40;
% Амплитуда первого сигнала:
a1 = 1;
% Амплитуда второго сигнала:
a2 = 0.5;
% Начало отсчетов времени:
t = 0:1/fd:tmax;
% Длительность:
fd2 = fd/2;

% Для создания новых массивов:
signal1 = a1 * sin(2*pi*t*f1);
signal2 = a2 * sin(2*pi*t*f2);
% График первого сигнала:
plot(signal1,[0,1]);
title('Signal 1');
% Сохранение графика в файл в каталоге signal:
print('signal/signal1.png');

% Построение спектра первого сигнала 1:
spectrel1 = abs(fft(signal1,fd));
spectre2 = abs(fft(signal2,fd));
% Построение спектра второго сигнала 2:
% Использование графиком спектров сигналов:
plot(spectrel1,[0,1]);
hold on;
plot(spectre2,[0,1]);
hold off;
title('Spectre');
% Сохранение спектра в файл в каталоге signal:
print('spectre/spectre.png');
```

Рис. 14: Модификация кода

Результат

И посмотрим на результат (рис. (fig:015?)).

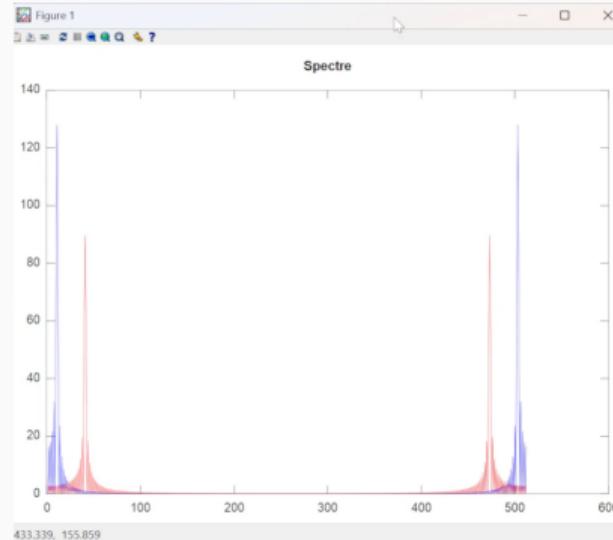
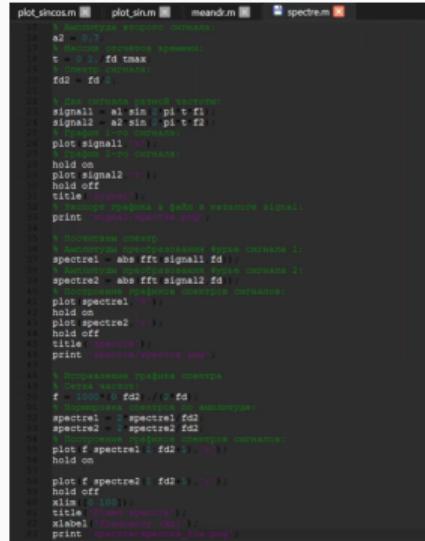


Рис. 15: Результат

КОД

Снова модифицируем код (рис. (fig:016?)).



```
plot_sincos.m plot_sinus.m meandr.m spectre.m
1 % Настройка параметров сигналов
2 a1 = 0.5;
3 % Настройка временных параметров
4 t1 = 3; fd_max
5 t2 = 10;
6 fd2 = fd_0;
7 % Генерация двух сигналов
8 signal1 = a1 * sin(2*pi*t1*fd);
9 signal2 = a2 * sin(2*pi*t2*fd2);
10 % Построение графиков
11 plot(signal1);
12 hold on
13 plot(signal2);
14 hold off
15 title('Сигналы');
16 % Сохранение графиков в файлы с наименованием signal1
17 print('signal1/signal1.png');
18 % Построение спектр
19 % Вычисление преобразования Фурье сигнала 1
20 spectrel1 = abs(fft(signal1,fd));
21 % Вычисление преобразования Фурье сигнала 2
22 spectrel2 = abs(fft(signal2,fd2));
23 % Построение графиков спектров сигналов
24 plot(spectrel1);
25 hold on
26 plot(spectrel2);
27 hold off
28 title('Спектры');
29 print('spectre/spectre.png');
30 % Построение графика спектра
31 % Старт частоты
32 f = 1000*(0:fd2)/fd;
33 % Вычисление спектра сигнала 1
34 spectrel1 = spectrel1(fd2);
35 spectrel2 = spectrel2(fd2);
36 % Вычисление спектра спектров сигналов
37 plot(f,spectrel1/fd2);
38 hold on
39 plot(f,spectrel2/fd2);
40 hold off
41 title('Спектр спектров');
42 xlabel('Frequency (Hz)');
43 print('spectre/spectre_fix.png');
```

Рис. 16: код

Результат

И посмотрим на результат (рис. (fig:017?)).

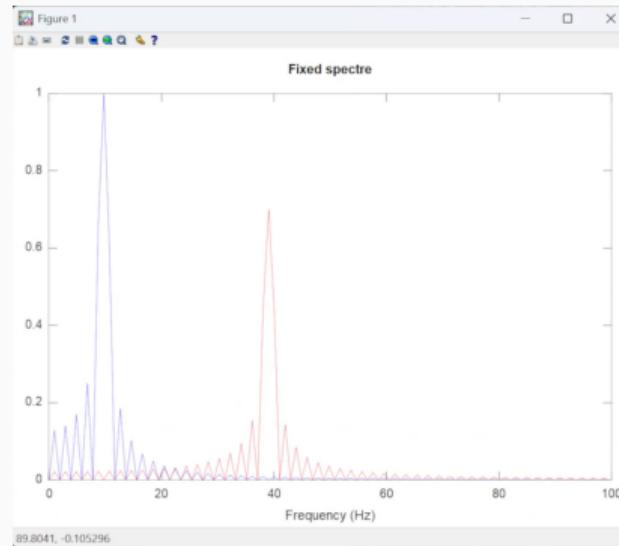


Рис. 17: Результат

Новый файл

Создадим новый файл (рис. (fig:018?)).

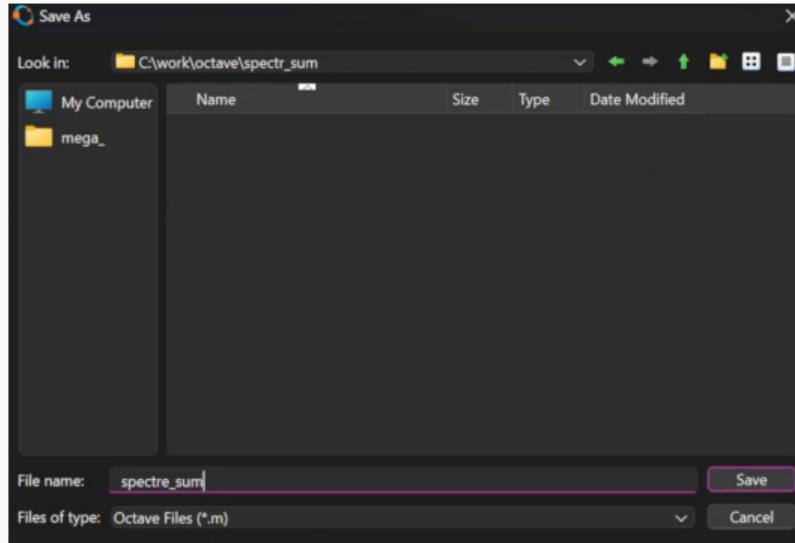


Рис. 18: Новый файл

Код

Напишем следующий код (рис. (fig:019?)).

```
plot_sincos.m plot_sin.m meandr.m spectre.m spectre_sum.m
1 % spectre_sum.m<br>
2 % Создание матричного signal и спектре для размещения графиков:<br>
3 mkdir('signals');<br>
4 mkdir('spectres');<br>
5 % Длина сигнала (n)<br>
6 tmax = 0.5;<br>
7 % Число дискретизаций (Nu) (количество точек):<br>
8 fd = 512;<br>
9 % Частота первого сигнала (Гц):<br>
10 f1 = 10;<br>
11 % Частота второго сигнала (Гц):<br>
12 f2 = 40;<br>
13 % Амплитуда первого сигнала:<br>
14 a1 = 1;<br>
15 % Амплитуда второго сигнала:<br>
16 a2 = 0.7;<br>
17 % Частота суммы:<br>
18 fd2 = fd/2;<br>
19 % Сумма двух сигналов (смесь) из разной частоты:<br>
20 % Максимум временн:<br>
21 t = 0:1(fd);<br>
22 signal1 = a1 sin(2 pi t f1);<br>
23 signal2 = a2 sin(2 pi t f2);<br>
24 signal = signal1 + signal2;<br>
25 plot(signal);<br>
26 title('signals');<br>
27 print('signals/signals_sum.png');<br>
28 % спектр суммы:<br>
29 % Вычисление спектральных Fourier-спектров:<br>
30 spectre = fft(signal,fd);<br>
31 % Нормированное спектро:<br>
32 f = 1000*(0:fd2-1)/fd;<br>
33 % Нормированная спектр по амплитуде:<br>
34 spectre = abs(sqrt(spectre.*conj(spectre)) ./ fd2);<br>
35 % Нормированное спектр спектра спектра:<br>
36 plot(f,spectre(1:fd2-1));<br>
37 title('spectres');<br>
38 xlabel('Частота (Гц)');<br>
39 ylabel('Спектр (МВт)');<br>
40 print('spectres/spectre_sum.png');
```

Рис. 19: Код

Результат

Посмотрим на результат (рис. (fig:020?)).

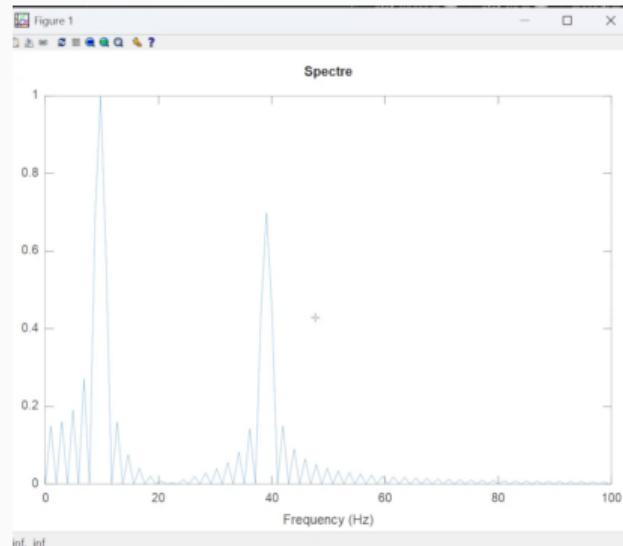


Рис. 20: Результат

Новый файл

Создадим новый файл (рис. (fig:021?)).

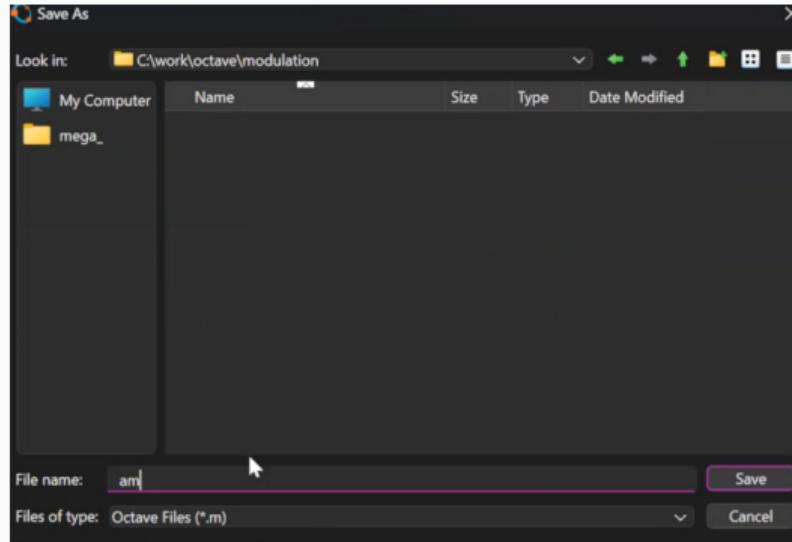
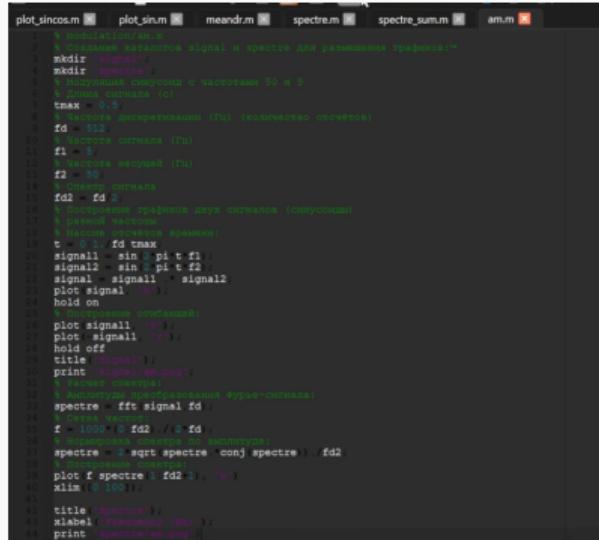


Рис. 21: Новый файл

Написание кода

И напишем в нём следующее (рис. (fig:022?)).



```
plot_sincos.m plot_sinc.m meandr.m spectre.m spectre_sum.m am.m
1 % создание вспомогательного signal и хранение для дальнейшего графиков
2 % mkdir 'plots';
3 % mkdir 'spectres';
4 % создание синусов с частотами 50 и 60
5 % длина сигнала (с)
6 tmax = 0.5;
7 % частота дискретизации (Гц) (количество отсчетов)
8 fd = 512;
9 % частота сигнала (Гц)
10 f1 = 5;
11 % частота высущий (Гц)
12 f2 = 60;
13 % фаза сигнала
14 fd2 = fd;
15 % создание временного профайла двух сигналов (суммусы)
16 % базовой частоты
17 % нахождение отнбеса времени
18 t = 0:1(fd*tmax);
19 signal1 = sin(2*pi*t*f1);
20 signal2 = sin(2*pi*t*f2);
21 signal = signal1 + signal2;
22 plot(signal);
23 hold on;
24 % воспроизведение отрываний
25 plot(signal1, 'r');
26 plot(signal2, 'b');
27 hold off;
28 title('Signal');
29 print('signal.png');
30 % вспомогательный
31 % Амплитудно-частотная характеристика-сигнала
32 spectre = fft(signal,fd);
33 f = 1000*(0:fd2)/fd;
34 % Амплитуда частоты по ампнтууму
35 spectre = 2*sqrt(spectre.*conj(spectre))/fd2;
36 % Амплитудно-частотная характеристика
37 plot(f,spectre(1:fd2), 'r');
38 xlim([0 10]);
39
40 title('Spectre');
41 xlabel('Frequency (Hz)');
42 print('spectre.png');
```

Рис. 22: Написание кода

результат

Посмотрим на результат (рис. (fig:023?)).

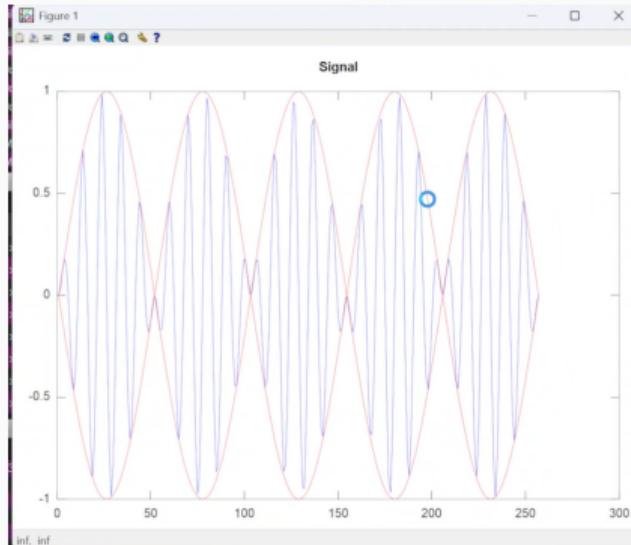


Рис. 23: результат

Второй вывод

И на второй вывод (рис. (fig:024?)).

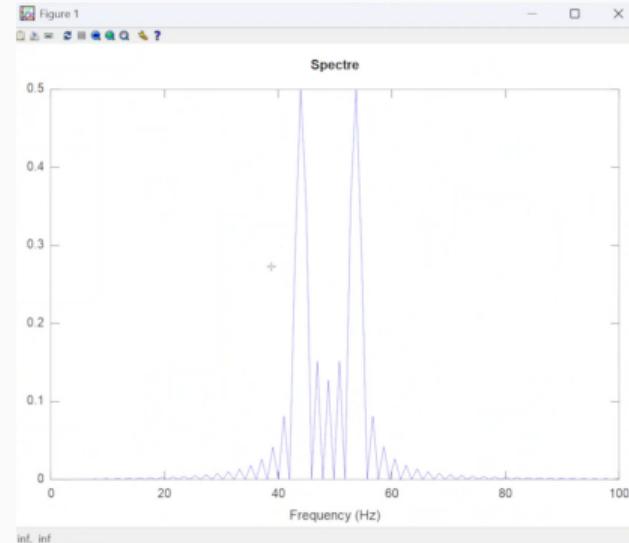


Рис. 24: Второй вывод

файлы

Создадим следующие файлы (рис. (fig:025?)).

The screenshot shows a Windows File Explorer window with the following directory path: C:\ Этот компьютер > Windows (C:) > work > octave > coding. The file list is as follows:

Имя	Дата изменения	Тип	Размер
ami.m	13.09.2025 14:45	Файл "M"	0 КБ
bipolarrrz.m	13.09.2025 14:45	Файл "M"	0 КБ
bipolarzz.m	13.09.2025 14:46	Файл "M"	0 КБ
<input checked="" type="checkbox"/> calcspectre.m	13.09.2025 14:46	Файл "M"	0 КБ
diffmanc.m	13.09.2025 14:46	Файл "M"	0 КБ
main.m	13.09.2025 14:45	Файл "M"	0 КБ
manchester.m	13.09.2025 14:46	Файл "M"	0 КБ
maptowave.m	13.09.2025 14:45	Файл "M"	0 КБ
unipolar.m	13.09.2025 14:45	Файл "M"	0 КБ

Рис. 25: файлы

signal

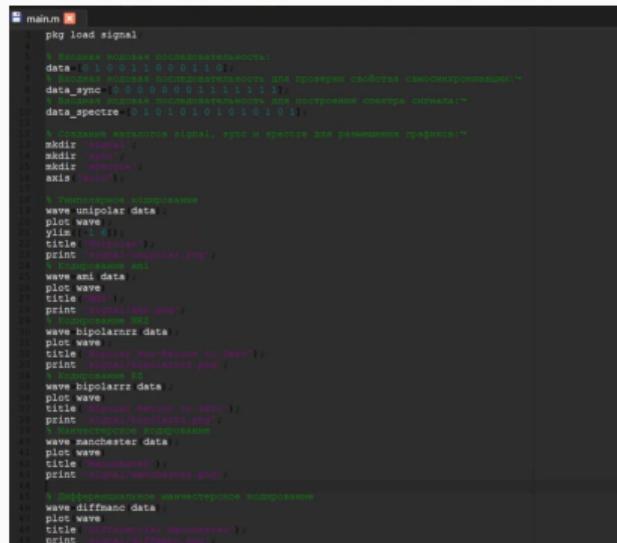
Убедимся, что установлен пакет signal (рис. (fig:026?)).

>> pkg list	Package Name	Version	Installation directory
	audio	2.0.5	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\audio-2.0.9
	biosig	3.9.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\biosig-3.9.0
	cfitso	0.0.7	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\cfitso-0.0.7
	coder	1.10.1	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\coder-1.10.1
	communications	1.2.7	...\\mingw64\\share\\octave\\packages\\communications-1.2.7
	control	4.1.2	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\control-4.1.2
	data-smoothing	1.3.0	...\\mingw64\\share\\octave\\packages\\data-smoothing-1.3.0
	database	2.4.4	...\\mingw64\\share\\octave\\packages\\database-2.4.4
	datatime	1.0.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\datatime-1.0
	dicom	0.6.1	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\dicom-0.6.1
	financial	0.5.3	...\\mingw64\\share\\octave\\packages\\financial-0.5.3
	fl-core	1.0.2	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\fl-core-1.0.2
	fuzzy-logic-toolkit	0.6.2	...\\mingw64\\share\\octave\\packages\\fuzzy-logic-toolkit-0.6.2
	ga	0.10.4	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\ga-0.10.4
	general	2.1.3	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\general-2.1.3
	generate_html	0.3.3	...\\mingw64\\share\\octave\\packages\\generate_html-0.3.3
	geometry	4.1.0	...\\mingw64\\share\\octave\\packages\\geometry-4.1.0
	image	2.1.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\image-2.1.0
	image-acquisition	2.16.1	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\image-acquisition-2.16.1
	instrument-control	0.3.3	...\\mingw64\\share\\octave\\packages\\instrument-control-0.3.3
	interval	3.2.1	...\\mingw64\\share\\octave\\packages\\interval-3.2.1
	io	2.7.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\io-2.7.0
	linear-algebra	2.2.3	...\\mingw64\\share\\octave\\packages\\linear-algebra-2.2.3
	lssa	0.1.4	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\lssa-0.1.4
	ltfat	2.6.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\ltfat-2.6.0
	mapping	1.0.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\mapping-1.0.0
	matgeom	1.2.4	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\matgeom-1.2.4
	miscellaneous	1.3.1	...\\mingw64\\share\\octave\\packages\\miscellaneous-1.3.1
	mqtt	0.0.5	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\mqtt-0.0.5
	nan	3.7.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\nan-3.7.0
	netcdf	1.0.18	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\netcdf-1.0.18
	nurbs	1.4.4	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\nurbs-1.4.4
	ocs	0.1.5	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\ocs-0.1.5
	octproj	3.0.0	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\octproj-3.0.0
	optim	3.0.2	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\optim-3.0.2
	optiminterp	0.3.7	...\\mingw64\\share\\octave\\packages\\optiminterp-0.3.7
	parallel	4.0.2	...\\mingw64\\share\\octave\\packages\\parallel-4.0.2
	quaternion	2.4.0	...\\mingw64\\share\\octave\\packages\\quaternion-2.4.0
	queueing	1.2.8	...\\mingw64\\share\\octave\\packages\\queueing-1.2.8
	signal	1.4.6	...\\Octave-10.2.0\\mingw64\\share\\octave\\packages\\signal-1.4.6

Рис. 26: signal

main

Напишем main функцию (рис. (fig:027?)).



```
pkg load signal

% Входные данные последовательности:
data=[0 1 0 0 1 1 0 0 0 1 1 0];
% Входные данные последовательности для проверки свойств амплитудно-импульсного кодирования:
data_sync=[0 0 0 0 0 0 1 1 1 1 1 1];
% Входные данные для проверки распространения сигнала:
data_spectre=[0 1 0 1 0 1 0 1 0 1 0 1 0];

% Установка путей к папкам, куда и в何处 для размещения графиков:
mkdir('signals');
mkdir('plots');
mkdir('spectre');
axis([0 100 0 1]);

% Проверка амплитудного кодирования
wave.unipolar data;
plot wave;
ylin([1 4]);
title('Unipolar');
print('signals/unipolar.png');

wave.uni data;
plot wave;
title('Uni');
print('signals/uni.png');

wave.bipolarrr data;
plot wave;
title('Bipolar Non-Norm to Zero');
print('signals/bipolarrr.png');

wave.bipolarz data;
plot wave;
title('Bipolar Normal to Zero');
print('signals/bipolarz.png');

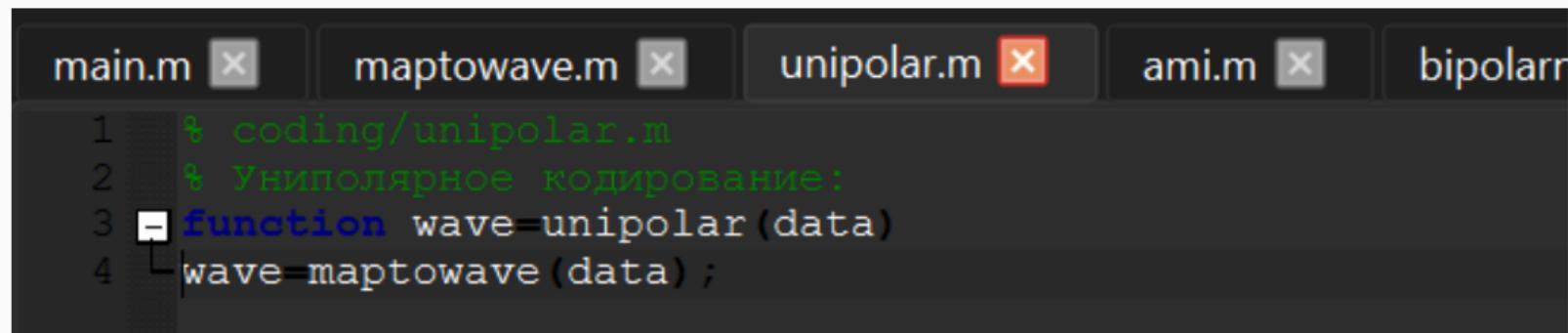
wave.manchester data;
plot wave;
title('Manchester');
print('signals/manchester.png');

% Дифференциальное曼彻斯特ское кодирование
wave.diffmanc data;
plot wave;
title('Differential Manchester');
print('signals/diffmanc.png');
```

Рис. 27: main

unipolar

И коды для различных алгоритмов (рис. (fig:028?)).



```
main.m x maptowave.m x unipolar.m x ami.m x bipolar.m x
1 % coding/unipolar.m
2 % Униполярное кодирование:
3 function wave=unipolar(data)
4 wave=maptowave(data);
```

Рис. 28: unipolar

ами (рис. (fig:029?)).

The screenshot shows a MATLAB workspace window. At the top, there are tabs for several files: main.m, maptowave.m, unipolar.m, ami.m (which is currently selected and has a red border), bipolarnrz.m, and bipolar. Below the tabs, the code for ami.m is displayed:

```
1 % coding/ami.m
2 % Кодирование АМI:
3 function wave=ami(data)
4 am=mod(1:length(data(data==1)),2);
5 am(am==0)=-1;
6 data(data==1)=am;
7 wave=maptowave(data);
```

Рис. 29: ami

bipolarnrz

bipolarnrz (рис. (fig:030?)).

The screenshot shows a MATLAB workspace window with several files listed in the tabs at the top: main.m, maptowave.m, unipolar.m, ami.m, bipolarnrz.m, and bip. The bipolarnrz.m file is currently selected and its contents are displayed in the editor area below. The code is as follows:

```
1 % coding/bipolarnrz.m
2 % Кодирование NRZ:
3 function wave=bipolarnrz(data)
4 data(data==0)=-1;
5 wave=maptowave(data);
```

Рис. 30: bipolarnrz

bipolarrz

bipolarrz (рис. (fig:031?)).

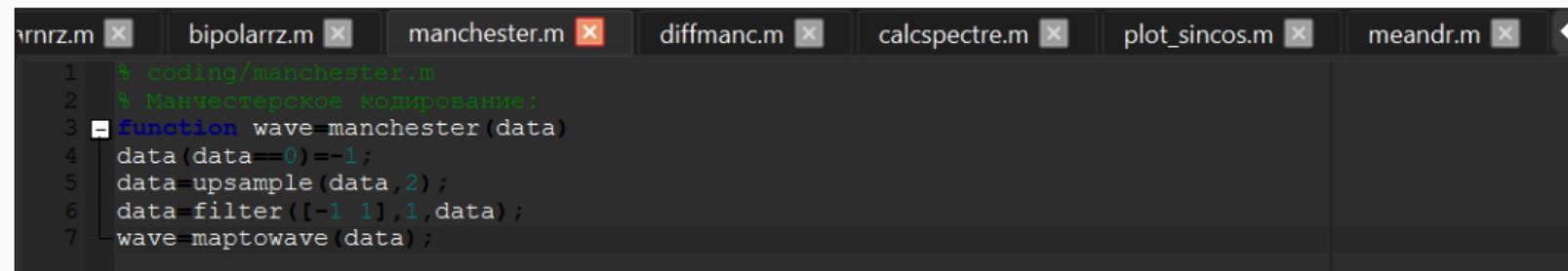
The screenshot shows a MATLAB workspace with several open files: main.m, maptowave.m, unipolar.m, ami.m, bipolarnrz.m, bipolarrz.m (which is the active file), and manchest. The bipolarrz.m file contains the following code:

```
1 % coding/bipolarrz.m
2 % Кодирование РЗ:
3 function wave=bipolarrz(data)
4 data(data==0)=-1;
5 data=upsample(data,2);
6 wave=maptowave(data);
```

Рис. 31: bipolarrz

manchester

manchester (рис. (fig:032?)).



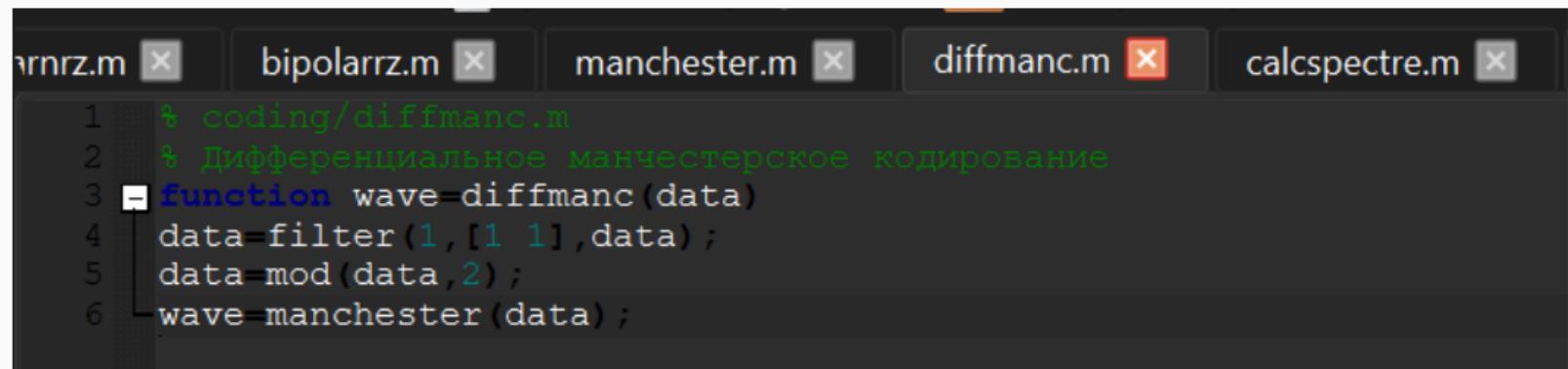
The screenshot shows a MATLAB interface with multiple tabs at the top: 'rnrz.m', 'bipolarrz.m', 'manchester.m' (which is currently active), 'diffmanc.m', 'calcspectre.m', 'plot_sincos.m', and 'meandr.m'. The 'manchester.m' tab has a red 'X' icon. Below the tabs, the code for 'manchester.m' is displayed:

```
1 % coding/manchester.m
2 % Манчестерское кодирование:
3 function wave=manchester(data)
4 data(data==0)=-1;
5 data=upsample(data,2);
6 data=filter([-1 1],1,data);
7 wave=maptowave(data);
```

Рис. 32: manchester

diffmanc

diffmanc (рис. (fig:033?)).



The screenshot shows a MATLAB workspace window with several files listed in the tabs at the top: arrnrz.m, bipolarrz.m, manchester.m, diffmanc.m (which is currently selected), and calcspectre.m. Below the tabs, the code for diffmanc.m is displayed in a text editor:

```
1 % coding/diffmanc.m
2 % Дифференциальное манчестерское кодирование
3 function wave=diffmanc(data)
4 data=filter(1,[1 1],data);
5 data=mod(data,2);
6 wave=manchester(data);
```

Рис. 33: diffmanc

calcspectre

calcspectre (рис. (fig:034?)).

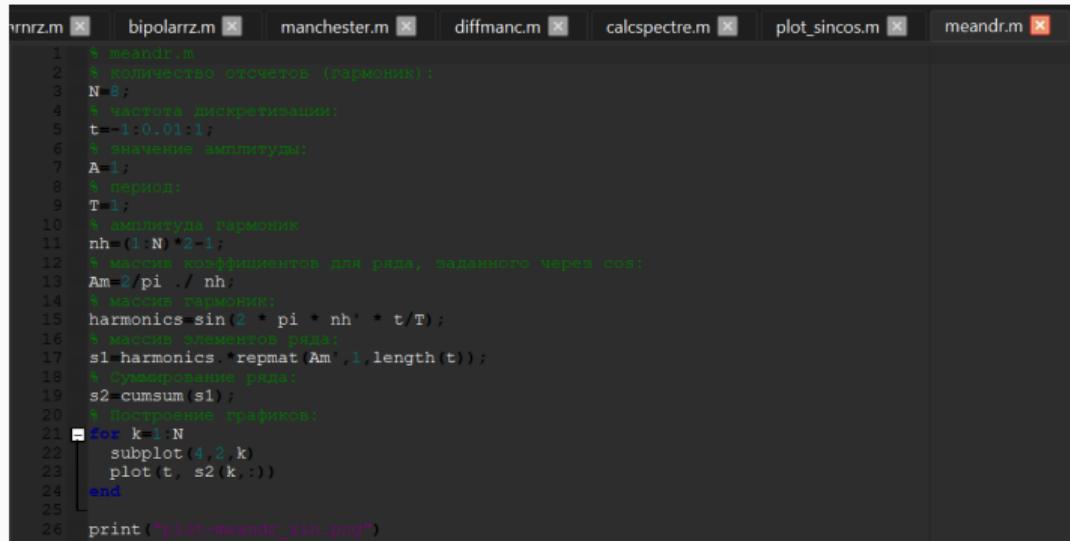
The screenshot shows a MATLAB interface with several tabs at the top: 'varnrz.m', 'bipolarzz.m', 'manchester.m', 'diffmanc.m', 'calcspectre.m' (which is the active tab), and 'p'. Below the tabs, the code for 'calcspectre.m' is displayed:

```
1 % calcspectre.m
2 % функция построения спектра сигнала:
3 function spectre = calcspectre(wave)
4 % Частота дискретизации (Гц):
5 Fd = 512;
6 Fd2 = Fd/2;
7 Fd3 = Fd/2 + 1;
8 X = fft(wave,Fd);
9 spectre = X.*conj(X)/Fd;
10 f = 1000*(0:Fd2)/Fd;
11 plot(f,spectre(1:Fd3));
12 xlabel('Frequency (Hz)');
```

Рис. 34: calcspectre

meandr

meandr (рис. (fig:035?)).



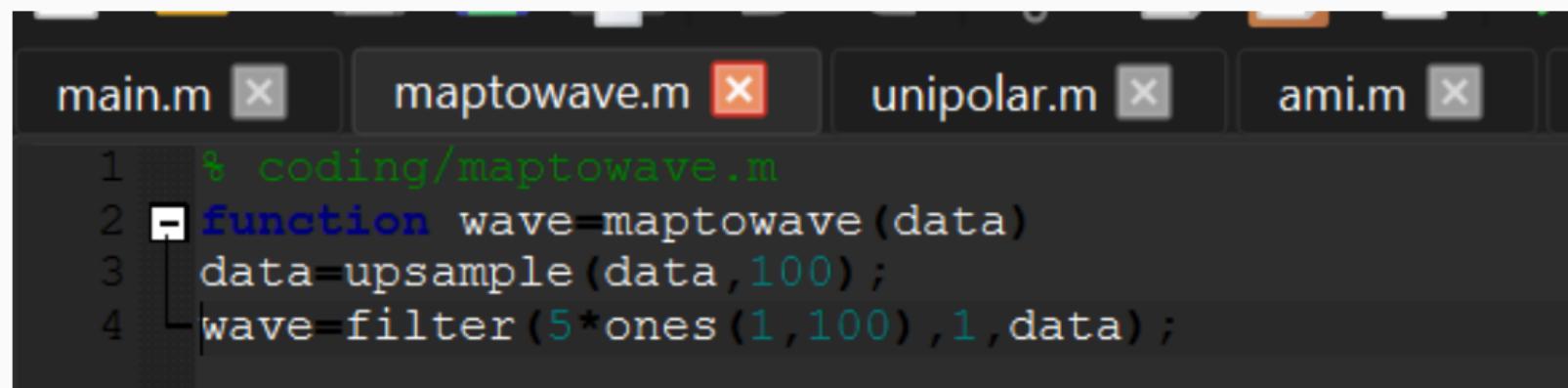
The screenshot shows a MATLAB interface with several tabs open at the top: vnrz.m, bipolarrz.m, manchester.m, diffmanc.m, calcspectre.m, plot_sincos.m, and meandr.m. The 'meandr.m' tab is active. Below the tabs, the code for 'meandr.m' is displayed:

```
1 % meandr.m
2 % количество отсчетов (гармоник):
3 N=8;
4 % частота дискретизации:
5 t=-1:0.01:1;
6 % значение амплитуды:
7 A=1;
8 % период:
9 T=1;
10 % амплитуда гармоник:
11 nh=(1:N *2-1;
12 % массив коэффициентов для ряда, заданного через cos:
13 Am=2/pi ./ nh;
14 % массив гармоник:
harmonics=sin(2 * pi * nh' * t/T);
16 % массив элементов ряда:
s1=harmonics.*kernmat(Am',1,length(t));
18 % суммирование ряда:
s2=cumsum(s1);
20 % Построение графиков:
21 for k=1:N
22 subplot(4,2,k)
23 plot(t, s2(k,:))
24 end
25
26 print("plot-meandr_sin.png")
```

Рис. 35: meandr

maptowave

maptowave (рис. (fig:036?)).



```
1 % coding/maptowave.m
2 function wave=maptowave(data)
3 data=upsample(data,100);
4 wave=filter(5*ones(1,100),1,data);
```

Рис. 36: maptowave

результат

И посмотрим на итоговый результат (рис. (fig:037?)).

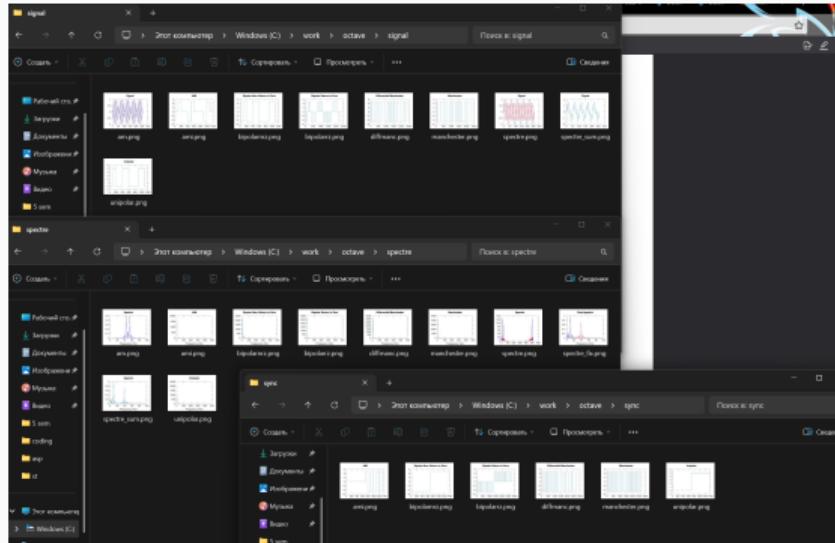


Рис. 37: результат

Выводы

в результате выполнения работы была установлена система