TRANSPORTA UN SAKARU INSTITŪTS



**Домашняя работа по дисциплине**

«Численные методы и прикладное программирование»

Выполнил: Анатолий Миронов

Группа: **4501BN**

Рига – 2019

# Contents

1. Решение системы линейных уравнений методом исключения Гаусса ..............................................3
2. Интерполяция и аппроксимация............................................................................................................5

Полином Ньютона 3-го порядка ............................................................................................................6

Аппроксимация .......................................................................................................................................6

1. Интегрирование и дифференцирование ...............................................................................................9
2. Решить нелинейное уравнение: ...........................................................................................................14

Метод бисекции ....................................................................................................................................14

Метод хорд ............................................................................................................................................14

Метод Ньютона .....................................................................................................................................14

Метод простых итераций .....................................................................................................................15

I-1 Решение системы ОДУ (модель Ресслера): .....................................................................................16

II -2 Минимизировать функцию при наличии ограничений. ...............................................................23

# Решение системы линейных уравнений методом исключения Гаусса

Ng = 28 Ns=7

* 1. Получение нулей под главной диагональю

**2 \*e1+e2**\*

Матрица приведена к треугольному виду:

* 1. Обратный ход гаусса. Находим неизвестные начиная от 𝑥3:

𝑥3 =

* 1. Исходя из последнего уравнения, 𝑥2 и 𝑥3 должны быть равны. Можно проверить:

𝑥2 + \*0.066093=

32𝑥1 – 3\*+ 4 \*= 3=

* 1. Проверка с исходной системой:

1.6. Невязка

Δ𝑏1 = 3 − 2.999999= 0.000001

Δ𝑏2 = 1 − 0.999999= 0.000001

Δ𝑏3 = 0 + 0.000003 = 0.000003

|Δ𝑏𝑖|

𝜎(𝑏𝑖) = ∗ 100%

𝑏𝑖

𝜎(b1)=

𝜎(b2)=

## 2. Интерполяция и аппроксимация

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝑋 | -1 | 28 | 6 | 10 |
| 𝑌 | 1 | 2 | 8 | -2 |

1. Интерполировать функцию 𝑦(𝑥) полиномом Лагранжа 3-го порядка 𝐿3(𝑥). Выполнить проверку правильности интерполяции по всем точкам.
2. Интерполировать функцию 𝑦(𝑥) полиномом Ньютона 3-го порядка 𝐿3(𝑥).
3. Аппроксимировать функцию 𝑦(𝑥) по методу наименьших квадратов полиномом 2-го порядка 𝜑2(𝑥).
4. Построить графики интерполяции 𝐿3(𝑥) и аппроксимации 𝜑2(𝑥) на одном рисунке в интервале 𝑥 ∈ [𝑥𝑚𝑖𝑛,𝑥𝑚𝑎𝑥] из таблицы и отметить на поле графика заданные табличные точки.

### 2.1 Полином Лагранжа

Основные уравнения:

где

Проверка:

Проверка:

|  |  |
| --- | --- |
| **X** | **Y\*** |
| -1 | 1.000001 |
| 28 |  |
| 6 |  |
| 10 |  |

### 2.2 Полином Ньютона 3-го порядка

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝑋 | 𝑌 | Δy | Δ2𝑦 | Δ3𝑦 |
| -1 | 1 | 0.034483 | −0.043887 | 0.015238 |
| 28 | 2 | -0.272727 | 0.123737 | — |
| 6 | 8 | −2.5 | — | — |
| 10 | -2 | — | — | — |

Δ𝑦0 ==0.034483

Δ𝑦1 ==-0.272727

Δ𝑦2 ==-2.5

Δ2𝑦0==-=−0.043887

Δ2𝑦1===0.123737

Δ3𝑦0===0.015238

𝑁3(𝑥) = 𝑎0 + 𝑎1(𝑥 − 𝑥0) + 𝑎2(𝑥 − 𝑥0)(𝑥 − 𝑥1) + 𝑎3(𝑥 − 𝑥0)(𝑥 − 𝑥1)(𝑥 − 𝑥2), где:

𝑎0 = 𝑦0, 𝑎1 = Δ𝑦0,𝑎2 = Δ2𝑦0,𝑎3 = Δ3𝑦0

𝑁3(𝑥) = 1 + 0.034483 (𝑥 + 1) -0.043887 (𝑥 + 1)(𝑥 − 28) + 0.015238 (𝑥 + 1)(𝑥 − 28)(𝑥 − 6) =

= 1 + 0.034483 x+0.034483 - 0.043887 (𝑥2 − 27𝑥 − 28) + 0.015238 (𝑥2 − 27𝑥 − 28)(𝑥 − 6) =

= 1. 034483+ 0.034483 x - 0.043887 (𝑥2 − 27𝑥 − 28) + 0.015238 (𝑥3 − 33𝑥2 – 134x+ 168) =

= 1. 034483+ 0.034483 x -0.043887x2+1.184949x +1.228836‬+0.015238x3 -0.502854x2 +2.041892x + 2.559984 =**0.015238x3--0.546741x2 -3.261324‬‬x + 4.823303**

И для сравнения интерполирующая функция полученная полиномом Лагранжа:

### 2.3 Аппроксимация

𝜑1(𝑥) = 𝑎0 + 𝑎1𝑥 + 𝑎2𝑥2

𝑛 = 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  | -1 | 1 | 1 | -1 | 1 | -1 | 1 |
|  | 6 | 8 | 36 | 216 | 1296 | 48 | 288 |
|  | 10 | -2 | 100 | 1000 | 10000 | -20 | -200 |
|  | 28 | 2 | 784 | 21952 | 614656 | 56 | 1568 |
| Сумма | **43** | **9** | **921** | **23167** | **625953** | **83** | **1657** |

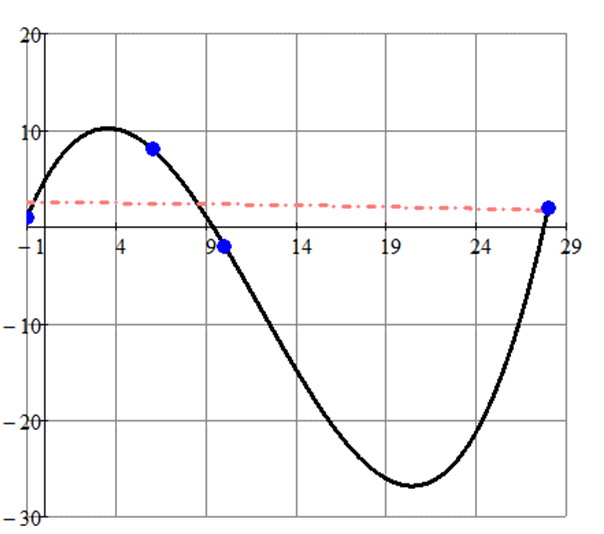
*Итоговый полином*

Проверка в Matlab:  
>> y=[1 2 8 -2];

>> x=[-1 28 6 10];

>> p=polyfit(x,y,2)   
p= -0.0006 -0.0131 2.5252

### 2.4 График



## Интегрирование и дифференцирование

|  |  |  |  |
| --- | --- | --- | --- |
| Номер варианта | Вид функции | Границы интервала | |
| *a* | *b* |
| 7 |  | 1 | 4 |

Вычислить определённый интеграл

на интервале , разделяя интервал на частей с шагом :

* + методом прямоугольников;
  + методом трапеций;
  + методом Симпсона.
  + С помощью квадратурной формулы Гаусса 3-го порядка
  1. сравнить полученные результаты

Вычислить производную по методу центральных разностей 𝑓′(𝑥) и интеграл с переменным 𝑥 верхним пределом

по методу трапеции, выбирая шаг ℎ. Результаты занести в таблицу.

Выбрать соответствующие масштабы и построить графики функций, 𝑓(𝑥), 𝑓′(𝑥),𝐹(𝑥) на одном рисунке в интервале 𝑥 ∈ [𝑎; 𝑏].

### 3.1 Вычисление определённого интеграла

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **x** | **f(x)** | **Si прям.** | **Si трап** | **Si Симпсон.** | **f`центр.разн** | **F(Xi)** |
| 1 | 8.0 | 4.8 |  |  |  |  |
| 1.6 | 44.572189 | 26.743313 | 15.771657 | 39.970165 | 171.177410 | 15.771657 |
| 2.2 | 213.412893 | 128.047736 | 77.395524 | 184.404703 | 751.0425675 | 93.167181 |
| 2.8 | 945.823270 | 567.493962 | 347.770849 | 799.601416 | 3154.912749 | 440.93803 |
| 3.4 | 3999.308192 | 2399.584916 | 1483.539439 | 3332.705604 | 12865.147275 | 1924.477469 |
| 4 | 16384.0 | 9830.4 | 6114.992458 | 2581.648749‬ |  | 8,039.469927 |
|  | Интеграл | 12952.269927 | 8039.469926 | 6938.330637 |  | 8,039.469927 |

### 3.2 сравнение результатов

Ответ, близкий к точному значению интеграла, дал метод Симпсона, но точнее является метод квадратуры Гаусса. Метод прямоугольников оказался менее точным.

### Квадратура Гаусса

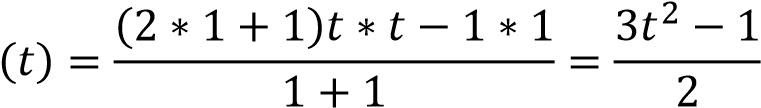
𝑏 𝑘

𝐼 = ∫ 𝑓(𝑥)𝑑𝑥 = ∑ 𝑐𝑚 ∗ 𝑥𝑚

𝑎 𝑚=1

𝑃0(𝑡) = 1

𝑃1(𝑡) = 𝑡

𝑃2

P3(t)=

t1=0 t2=-0,774507 t3=0,774507

P`3=7.5t2-1.5

Ci=

C1= C2=C3==0,555556

X=

X1=2.5 X2=3.6617605 X3=1.3382395

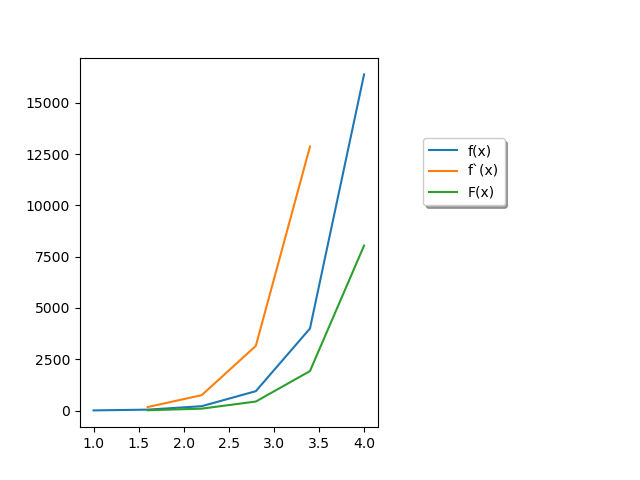
I = (C1f(x1)+C2f(x2)+C3(x3))\*

f(2.5)= 452.548339

f(3.6617605)= 7423.16609

f(1.3382395)= 21.631395

I=6807.40073



Сравнение полученных результатов:

|  |  |
| --- | --- |
| Прямоугольником | 12952.269927 |
| Трапецией | 8039.469926 |
| Симпсоном | 6938.330637 |
| Квадратурой Гаусса | 6807.40073 |

Наиболее точный результат дали метод квадратуры Гаусса и метод Симпсона. Метод трапеции так же близок к истинному значению.

**4. Решить нелинейное уравнение:**

𝑓(𝑥) = 𝑁𝑔

𝑥𝑒0.8𝑥 = 24

𝑓(𝑥) = 𝑥𝑒0.8𝑥 − 24 = 0

### Метод бисекции

Идея: делим исходный интервал пополам и находим значение на границах половинок. Если произведение значений функций на каждой половинке больше 0, тогда решение находится на второй половинке и сдвигаем левую границу интервала, в противном случае решение находится в текущей половинке и сдвигаем правую границу интервала. Повторяем итерацию.

𝑏−𝑎

Деление интервала: 𝑥𝑛𝑒𝑤 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№**  **итерации** | 𝒂 | 𝒃 | 𝒙𝒏𝒆𝒘 | 𝒇(𝒂) | 𝒇(𝒙𝒏𝒆𝒘) | 𝒇(𝒂) ∗ 𝒇(𝒙𝒏𝒆𝒘) |
| **1** | 2 | 3 | 2.5 | -14.093935 | -5.527359 | 77.90225 |
| **2** | 2.5 | 3 | 2.75 | -5.527359 | 0.818787 | -4.525731\* |
| **3** | 2.5 | 2.75 | 2.625 | -5.527359 | -2.563804 | 14.171067 |
| **4** | 2.625 | 2.75 | 2.6875 | -2.563804 | -0.928193 | 2.379705 |
| **5** | 2.6875 | 2.75 | 2.71875 | -0.928193 | -0.069059 | 0.0641 |
| **6** | 2.71875 | 2.75 | 2.734375 | -0.069059 | 0.371219 | -0.025636\* |
|  | 2.71875 | 2.734375 |  |  |  |  |
|  | Δ = | 0.015625 |  |  |  |  |

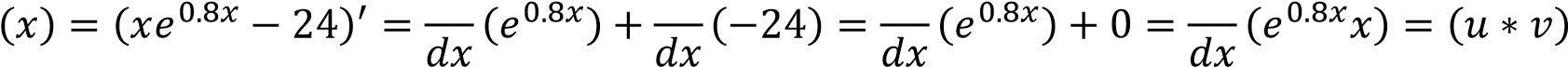
### Метод хорд

𝑏 − 𝑎

𝑥𝑛𝑒𝑤 = 𝑏 − 𝑓(𝑏) − 𝑓(𝑎) ∗ 𝑓(𝑏)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№**  **Итер.** | 𝒂 | 𝒃 | 𝒙𝒏𝒆𝒘 | 𝒇(𝒂) | 𝒇(𝒃) | 𝒇(𝒙𝒏𝒆𝒘) | 𝒇(𝒂) ∗ 𝒇(𝒙𝒏𝒆𝒘) |
| **1** | 2 | 3 | 2.608455 | -14.093935 | 9.069529 | -2.978987 | 41.985644 |
| **2** | 2.608455 | 3 | 2.705264 | -2.978997 | 9.069529 | -0.443277 | 1.320522 |
| **3** | 2.705264 | 3 | 2.718998 | -0.44329 | 9.069529 | -0.062114 | 0.027535 |
| **4** | 2.718998 | 3 | 2.72091 | -0.062127 | 9.069529 | -0.008631 | 0.000536 |
| **5** | 2.72091 | 3 | 2.721175 | -0.008625 | 9.069529 | -0.001197 | 0.00001 |
| **6** | 2.721175 | 3 | 2.721212 | -0.001201 | 9.069529 | -0.000167 | 2 ∗ 10−7 |
|  |  |  |  |  |  |  |  |
|  | Δ = |  |  |  |  |  |  |

### Метод Ньютона

𝑓′ 𝑑 𝑑 𝑑 𝑑 ′

= 𝑢′𝑣 + 𝑢𝑣′ = 𝑥′𝑒0.8𝑥 + 𝑥(𝑒0.8𝑥)′ = 𝑒0.8𝑥 + 𝑥𝑒0.8𝑥(0.8𝑥)′ = 𝑒0.8𝑥 + 0.8𝑥𝑒0.8𝑥 = 𝑒0.8𝑥(1 + 0.8𝑥).

𝑎 + 𝑏

𝑥0 =

2

𝑓(𝑥𝑘)

𝑥𝑘+1 = 𝑥𝑘 − 𝑓′(𝑥𝑘)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ итерации** | 𝒙 | 𝒇(𝒙) | 𝒇′(𝒙) | 𝒙𝒌+𝟏 |
| **1** | 2.5 | -5.52736 | 22.167168 | 2.74934893255957 |
| **2** | 2.74934893255957 | 0.799993 | 28.860309 | 2.72162952176629 |
| **3** | 2.72162952176629 | 0.01155 | 28.031729 | 2.72121796588936 |
| **4** | 2.72121796588936 | 2 ∗ 10−6 | 28.019583 | 2.7212178767881 |
| **5** | 2.7212178767881 | 8 ∗ 10−14 | 28.0195804 | 2.7212178767881 |
| **6** | - | - | - | - |
|  |  |  |  |  |

### Метод простых итераций

𝑓(𝑥) = 𝑥𝑒0.8𝑥 − 24

𝜑(𝑥) = 𝑓(𝑥)𝜉 + 𝑥 = 0.05(𝑥𝑒0.8𝑥 − 24) + 𝑥

В качестве корректирующего коэффициента опытным путём было выбрано значение 𝜉 = 0.05.

𝜑′(𝑥) = 𝜉𝑓′(𝑥) + 𝑥′ = 0.05(𝑒0.8𝑥(1 + 0.8𝑥)) + 1

𝑥𝑘+1 = 𝜑(𝑥𝑘)

Условие сходимости: |𝜑′(𝑥)| < 1 на интервале [𝑎; 𝑏].

|  |  |  |  |
| --- | --- | --- | --- |
| **№ итерации** | 𝒙 | 𝝋(𝒙) | 𝝋′(𝒙) |
| **1** | 2.5 | 2.77636798763367 | -0.1 |
| **2** | 2.77636798763367 | 2.69682149720422 | -0.48 |
| **3** | 2.69682149720422 | 2.73056530753639 | -0.36 |
| **4** | 2.73056530753639 | 2.71740517777619 | -0.41 |
| **5** | 2.71740517777619 | 2.72273599221887 | -0.39 |
| **6** | 2.72273599221887 | 2.72060744544847 | -0.4 |
|  |  |  |  |

## 5. Решить линейное дифференциальное уравнение второго порядка с постоянными коэффициентам

𝑦′′(𝑡) + 5𝑦′(𝑡) + 5𝑦(𝑡) = 24

𝑦(0) = 5

𝑦′(0) = 10

Задачу решить численным методом Рунге-Кутта 4-го порядка.

Строим характеристическое уравнение и находим его корни:

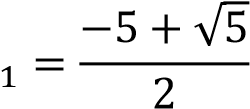
𝑝2 + 5𝑝 + 5 = 0

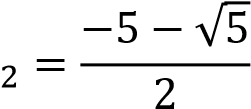
𝐷 

−𝑏 ± √𝐷

𝑝1,2 =

2𝑎

𝑝

𝑝

Вычисляем временной интервал:

1 1

𝜏 = = = 0.723

|𝑃𝑚𝑖𝑛| |−1.382|

𝑇 = 4𝜏 = 2.892

Вычисляем рекомендуемый шаг ℎ (для машинного счёта знаменатель устанавливаем в 200):

𝑇 2.892

ℎ = Δ𝑡 = = ≈ 0.014

Составляем уравнение:

𝑦′ = 𝑧, 𝑦(0) = 𝑦0

{𝑧′ = 24 − 5𝑧 − 5𝑦 𝑧(0) = 𝑦′(0) = 𝑦0′

### Решение в Matlab

Листинг файла kr.m:

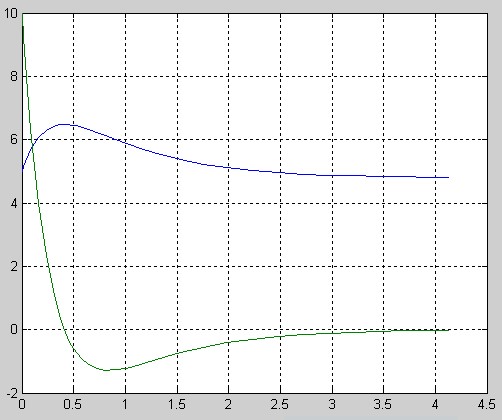
function f = kr(t, x) f = [x(2); 24 - 5.\*x(2)-5.\*x(1)]; решение дифференциального уравнения функцией ode23:

[t, x] = ode23('kr', [0 2.892], [5 10]);

**Результат:**

|  |  |  |
| --- | --- | --- |
| t |  | x |
| 0,0000 | 5,0000 | 10,0000 |
| 0,0157 | 5,1507 | 9,2247 |
| 0,0854 | 5,6883 | 6,3160 |
| 0,1602 | 6,0705 | 3,9969 |
| 0,2327 | 6,2981 | 2,3543 |
| 0,3007 | 6,4181 | 1,2230 |
| 0,3626 | 6,4691 | 0,4583 |
| 0,4038 | 6,4796 | 0,0608 |
| 0,4450 | 6,4752 | -0,2631 |
| 0,4804 | 6,4617 | -0,4911 |
| 0,5259 | 6,4338 | -0,7266 |
| 0,5809 | 6,3877 | -0,9394 |
| 0,6454 | 6,3213 | -1,1089 |
| 0,7196 | 6,2344 | -1,2230 |
| 0,8044 | 6,1280 | -1,2769 |
| 0,9012 | 6,0044 | -1,2713 |
| 1,0124 | 5,8661 | -1,2109 |
| 1,1439 | 5,7141 | -1,1007 |
| 1,3058 | 5,5487 | -0,9449 |
| 1,5311 | 5,3604 | -0,7361 |
| 1,8203 | 5,1814 | -0,5152 |
| 2,0549 | 5,0773 | -0,3783 |
| 2,3044 | 4,9970 | -0,2704 |
| 2,5495 | 4,9406 | -0,1935 |
| 2,7928 | 4,9004 | -0,1385 |
| 2,8920 | 4,8876 | -0,1209 |

**График:**



**II-8 Решение системы ОДУ (модель Ресслера):**

𝑥′ = −𝑦 − 𝑧

𝑦′ = 𝑥 + 𝐴𝑦

𝑧′ = 𝐵 + 𝑧(𝑥 − С)

С начальными условиями: 𝑥(0) = −0.7,𝑦(0) = −0.7,𝑧(0) = 1 на интервале времени 𝑡 = [0: 50], с параметром модели 𝐴 = 0.2,𝐵 = 0.2, 𝐶 = 5.7. Построить для каждого значения 𝐴 = 0.1, 0.2,0.35 графики 𝑥(𝑡), 𝑦(𝑡), 𝑧(𝑡) и фазовый портрет системы.

Листинг файла lorenc.m:

function f = lorenc(t,x)

f=[-x(2)-x(3);x(1)+0.2.\*x(2);0.2+x(3).\*(x(1)-5.7)];

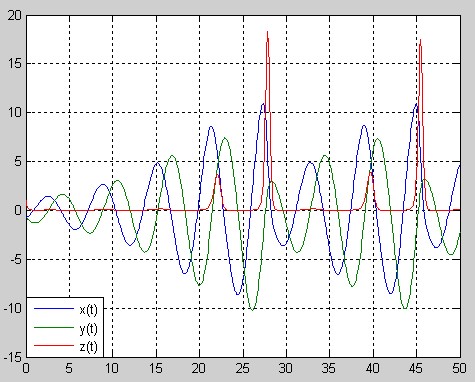
Мы заменили переменные 𝑥, 𝑦 и 𝑧 на 𝑥(1), 𝑥(2), 𝑥(3) соответственно и заменили коэффициенты

𝐴, 𝐵 и 𝐶 на заданные параметры модели. Далее, выполняем команды для решения дифференциального уравнения при помощи функции ode45:

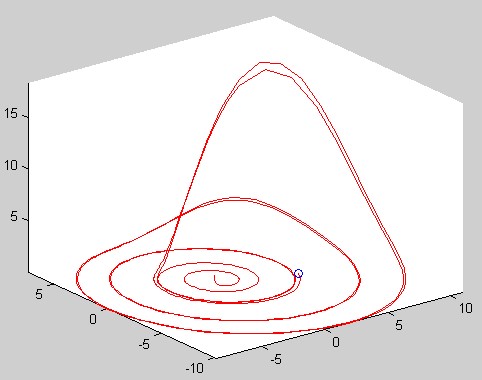
x0=[-0.7 -0.7 1]; % вектор-столбец начальных условий ts = [0 50]; % задаём временной интервал

[t,x]=ode23('lorenc',ts,x0); % решаем дифференциальное уравнение plot(t,x), grid on % строим график для A=0.2 comet3(x(:,1),x(:,2),x(:,3)) % строим фазовый портрет

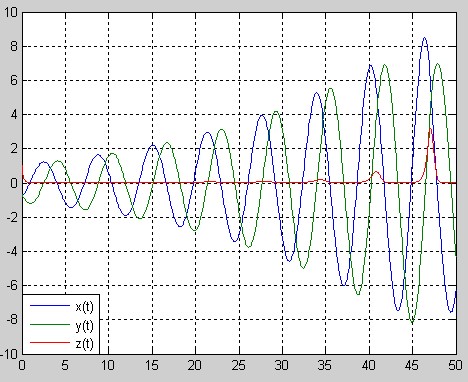
Графики:



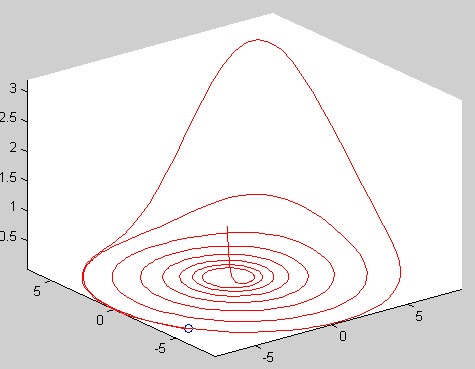
*Рис. 1 График для A=0.2*



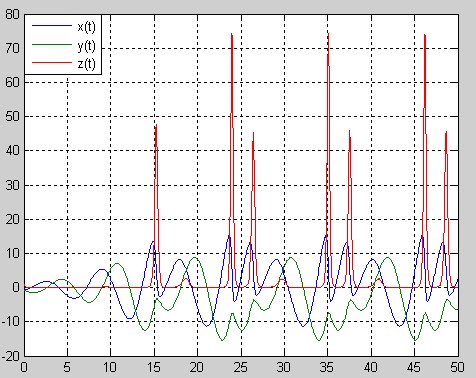
*Рис. 2 Фазовый портрет для A=0.2*



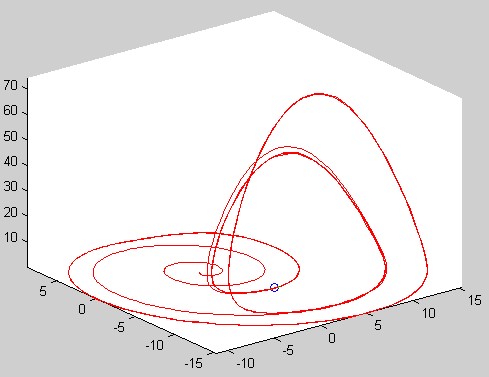
*Рис. 3 График для A=0.1*



*Рис. 4 Фазовый портрет для A=0.1*



*Рис. 5 График для A=0.35:*



*Рис. 6 Фазовый портрет для A=0.35*

**IV-2 Минимизировать функцию при наличии ограничений.**

Начальные ограничения: 𝑥0 = [1 1 1].

𝑓(𝑥1,𝑥2, 𝑥3) = 𝑥12 + 𝑥22 + 𝑥32 + 4𝑒−𝑥12−𝑥22−𝑥32 − 6𝑥1𝑥2𝑥3;

𝑥1 + 3𝑥2𝑥3 − 8 ≤ 0;

𝑥1 + 𝑥2 = 4 ⇒ 𝑥1 + 𝑥2 − 4 = 0;

𝑥3 ≥ 0 ⇒ −𝑥3 ≤ 0.

Листинг для fun.m:

function f = fun(x)

f = x(1).^2+x(2).^2+x(3).^2+4.\*exp(-x(1).^2-x(2).^2-x(3).^2)-6.\*x(1).\*x(2).\*x(3); листинг для ограничений mycon.m:

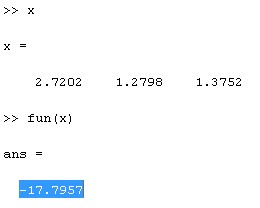
function [g,ce] = mycon(x) g(1) = x(1) + 3.\*x(2).\*x(3) - 8 g(2) = -x(3); ce = x(1) + x(2) - 4;

вызов функции fmincon:

x0 = [1 1 1];

x = fmincon('fun',x0,[],[],[],[],[],[],'mycon');

Результаты:



Проверка ограничений:

2.7202 + 3 ∗ 1.2798 ∗ 1.3752 ≤ 0 ⇒ ≈ 0 ⟹ Выполняется

{ 2,7202 + 1,2798 − 4 = 0 ⇒ Выполняется

1.3752 ≥ 0 ⇒ Выполняется

Минимум заданной функции будет достигаться при 𝑥1 = 2.7202, 𝑥2 = 1.2798,𝑥3 = 1.3752 и значение заданной функции при заданных параметрах будет равно -17.7957.