**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра КСУ**

отчет

**по идз №2**

**по дисциплине «ПРОЕКТИРОВАНИЕ**

**ОПТИМАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ»**

Тема: Решение задач оптимизации методом поиска

**Вариант 12**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4491 | Пономарев Д.А. |  |
| Преподаватель | Ветчинкин А.С. |  |

Санкт-Петербург

2018

**Решение дифференциальных уравнений**

**Задача 1.** **Исходные данные**

Статическая задача.

Определить глобальный максимум функции и исследовать поведение функции в районе экстремума.

Исходные данные заданы в таблице 1.

Таблица 1. Исходные данные к заданию

|  |  |
| --- | --- |
| Вариант | Функция |
| 12 |  |

**Задача 1.** **Аналитическое решение**

Для того, чтобы найти глобальный максимум, сначала найдем все точки экстремума искомой функции.

Для этого вычислим частные производные:

Получаем следующие точки экстремумов:

1. x=-2, y = 1

Очевидно, что это минимальное значение z. Это можно доказать, если представить уравнение в виде уравнения окружности:

1. Множество точек, полученных из уравнения

Преобразуем уравнение к уравнению окружности:

Очевидно здесь k >= 1.

Таким образом, количество локальных точек экстремума бесконечно.

Анализировать их все по одной бесполезно, поэтому проанализируем искомую функцию.

Искомая функция состоит из произведения двух множителей:

1. Экспоненциальный множитель . Его значение изменяется от бесконечности до нуля, притом большему значению z соответствует меньшее значение экспоненциальной составляющей. При этом минимальное значение z=0 (соответствует x=-2. y=1), как говорилось ранее.

2. Множитель = принимает значения от -1 до 1.

Из всего вышесказанного и с учетом того, что косинус – периодическая функция, получаем 3 «кандидата» на роль глобального максимума:

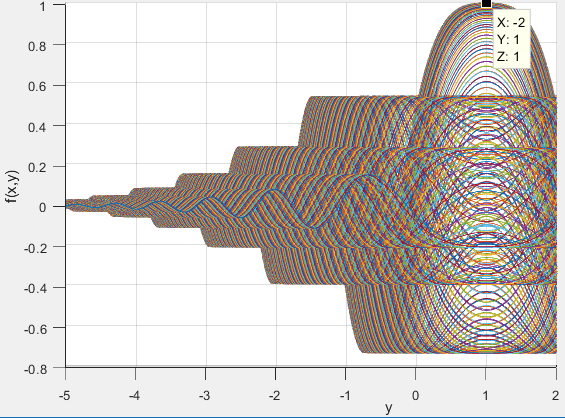
1. Точка x=-2, y=1, т.е. z=0

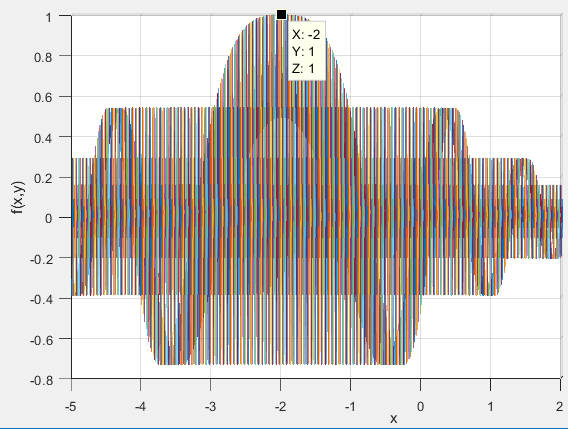
2. Множество точек

3. Множество точек

Методом подстановки данных значений получаем, что точка x=-2, y=1, т.е. z=0 – то глобальный максимум искомой функции.

Представим графики искомой функции f(x,y) на рисунках 1 и 2.





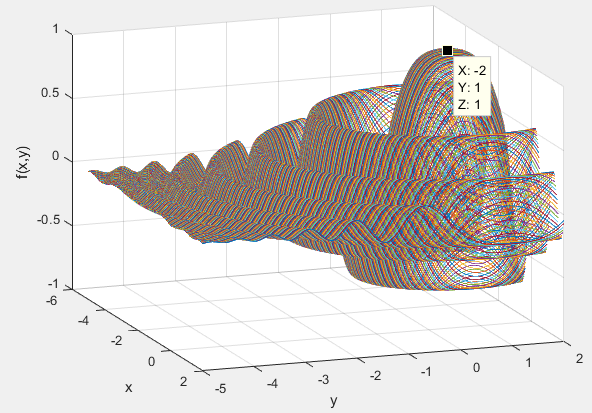


Рисунок 1 – Графики функции f(x,y)

Функция f(z) представлена на рисунке 2.

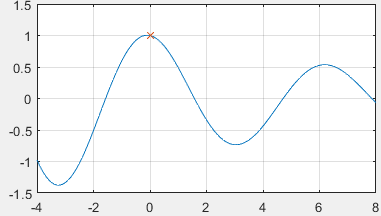


Рисунок 2 – График функции f(z)

Графики построены с помощью кода программы, представленного на рисунке 3.

|  |
| --- |
| %file idz2\_part1.m - main  clear; close all; clc;    % Analytical calculation  figure  [x, y] = meshgrid(-5 : 0.02 : 2);  f = exp(-0.1.\*(x.^2+y.^2+4.\*x-2.\*y+5)).\*cos(x.^2+y.^2+4.\*x-2.\*y+5);  plot3(x, y, f)  figure  z = -4 : 0.1 : 8;  plot(z, exp(-0.1.\*z).\*cos(z), 0, 1, 'x')  grid on |

Рисунок 3 – Код программы

**Задача 1. Численное решение в матлаб**

Численно решение в Matlab

Код в матлаб представлен на рисунке 4.

|  |
| --- |
| %file idz2\_part1.m - main  clear; close all; clc;  % Numerical calculation  x0 = [-1 1];  f\_global\_maximum\_arguments = fminsearch('calculate\_f', x0)  f\_global\_maximum\_value = -calculate\_f(f\_global\_maximum\_arguments) |
| function f = calculate\_f(argv)  x = argv(1); y = argv(2);  z = x.^2 + y.^2 + 4.\*x - 2.\*y + 5;  f = -1.\*exp(-0.1.\*z).\*cos(z);  end |

Рисунок 4. Код программы

Результаты выполнения представлены на рисунке 5. Значение аргументов (-2;1) соответствует аргументам х и у соответственно.

|  |
| --- |
| f\_global\_maximum\_arguments =  -2.0000 1.0000  f\_global\_maximum\_value =  1.0000 |

Рисунок 5. Результат выполнения программы

**Задача 2. Исходные данные**

Динамическая задача. Динамический объект



Входным воздействием на объект является функция времени, содержащая неизвестный параметр , величину которого необходимо определить из условия 

Функция  является кусочно-линейной, и её вид определяется следующей таблицей значений:

|  |  |
| --- | --- |
| Вариант | Функция |
| 12 |  |

**Задача 2. Аналитическое решение**

1. Определим уравнение функции u(t) в общем виде

Пусть функция u(t) имеет следующий вид:

где – единичная ступенчатая функция (функция Хевисайда), t0 = 0, t1 = 3.5, t2 = 9 – моменты времени, в которые определено значение функции .

Тогда функция имеет следующий вид, представленный на рисунке 5.

Данная функция полностью соответствует требуемым условиям.

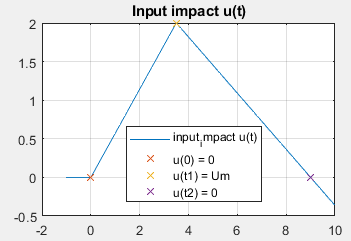


Рисунок 5. Функция входного воздействия u(t)

2. Найдем аналитическое решение функции x(t) при произвольном Um

Выполним преобразование Лапласа:

(1)

где

*–* начальное условие.

Подставим значения и в уравнение (1):

Выразим x(s):

Разложим второе слагаемое на простейшие:

Третье слагаемое раскладывается аналогично.

Выполним обратное преобразование Лапласа:

(2)

3. Аналитически найдем значение Um, при котором x(9) = 0.

Для этого подставим в уравнение (2) значения t = 9 и x = 9.

+

**Задача 2. Численное решение в Matlab**

Преобразуем исходную задачу к задаче поиска минимума.

Новая формулировка задачи заключается в том, что мы потребуем найти такое значение параметра входного воздействия, которое обеспечит минимальное значение |х(9)|. Такая постановка задачи позволяет использовать при решении поставленной задачи MATLAB – функцию FMINSEARCH. Кроме функции FMINSEARCH будет использована функция ODE45, которая предназначена для численного решения дифференциального уравнения.

Код основного скрипта представлен на рисунке 6. Данный скрипт выполняет 4 действия:

1. инициализирует исходные глобальные переменные;

2. выполняет запуск функции fminsearch;

3. строит график функции в соответствии с аналитическим решением;

4. строит график входного воздействия.

|  |
| --- |
| % Init data  clear; close all; clc;  global t\_start t\_top t\_end x0 U  x0 = 10; % x(0)  t\_start = 0; % the begin of heviside  t\_top = 3.5; % the top of heviside  t\_end = 9; % the end of heviside  U = 2; % heviside amplitude    % Numerical sulution - find Um  Um = fminsearch('fminsearch\_func', 0);  fprintf('Um = %f \n', Um);    % Analytical solution - Um perfect  t = t\_start : 0.1 : t\_end;  hold on; grid on;  plot(t, analytical\_func(t), 'rx', ...  t\_start, x0, ...  t\_end, 0)  title('Numberical and Analytical solutions')  xlabel('t, sec'); ylabel('x(t)');  legend('analytical solution', 'ode45 solution', 'start point', 'end point')    % Create heviside graph  figure;  plot(t, input\_impact(t), ... % input\_impact u(t)  0, 0, 'x', ... % u(t\_start) = 0  t\_top, U, 'x', ... % u(t\_top) = Um  t\_end, 0, 'x') % u(t\_end) = 0  grid on;  title('Input impact u(t)')  legend('input\_impact u(t)', 'u(t\_start) = 0', 'u(t\_top) = Um', 'u(t\_end) = 0') |

Рисунок 6 – Код основного скрипта

Код функции fminsearch\_func(), которая является аргументом функции fminsearch(), представлен на рисунке 7. В Данной функции выполняется расчет согласно новой формулировке задачи, которая представлена выше, с помощью ode45. Также строится текущий график.

|  |
| --- |
| function f = fminsearch\_func(U\_new)  % func\_for\_fminsearch(U) return abs( f(U, t = 9) )  % So, it should be used in fminsearch() function  global t\_start t\_end U x0  U = U\_new;  [t, x] = ode45('ode\_func', [t\_start t\_end], x0);  plot(t, x)  pause(0.01);  f = abs(x(end));  end |

Рисунок 7 – Код функции fminsearch\_func

Код функции ode\_func(), которая является аргументом функции ode45(), представлен на рисунке 8. В ней выполняется расчет правых частей уравнения.

|  |
| --- |
| function dxdt = ode\_func(t, x)  dxdt = -0.5.\*x + input\_impact(t);  end |

Рисунок 8 – Код функции ode\_func

Входное воздействия есть сумма функций Хевисайда. Чтобы не загромождать основной код формулами, расчет входного воздействия был вынесен в отдельную функцию - input\_impact(), код которой представлен на рисунке 9. Для того, чтобы часть кода программы можно было использовать в других задачах и для лучшей читаемости кода, функция Хевисайда в общем виде была так же вынесена в отдельную функцию Heaviside(), код которой представлен на рисунке 10.

|  |
| --- |
| function f = input\_impact(t)  global t\_start t\_top t\_end U  f = zeros(1, length(t));  for count = 1:length(t)  f(count) = Heaviside(t(count),t\_start) \* t(count) \* U/t\_top + ...  + Heaviside(t(count),t\_top) \* (t(count) - t\_top) \* (-U/t\_top - U/(t\_end-t\_top));  end  end |

Рисунок 9 – Код функции input\_impact

|  |
| --- |
| function f = Heaviside(t, t0)  % Heaviside(t, t0) is 0 for t < t0 and 1 for t > t0  f = zeros(1, length(t));  for count = 1:length(t)  if t(count) > t0  f(count) = 1;  else  f(count) = 0;  end  end |

Рисунок 10 – Код функции Heaviside

Функция расчета по аналитически полученной формуле выполняется в функции analytical\_func(), код которой представлен на рисунке 11.

|  |
| --- |
| function f = analytical\_func(t)  global U t\_start t\_top t\_end  f = 10\*exp(-0.5.\*t) ...  + Heaviside(t, t\_start).\*U/t\_top.\*(2.\*t+4.\*exp(-0.5.\*t)-4) ...  + Heaviside(t, t\_top).\*U.\*(-1./t\_top - 1./(t\_end-t\_top)) .\* (2.\*(t-t\_top) + 4.\*exp(-0.5.\*(t-t\_top)) - 4);  end |

Рисунок 11 – Код функции analytical\_func

Решения, полученные с помощью ode45 и fminsearch и аналитических расчетов, представлены на рисунке 12.

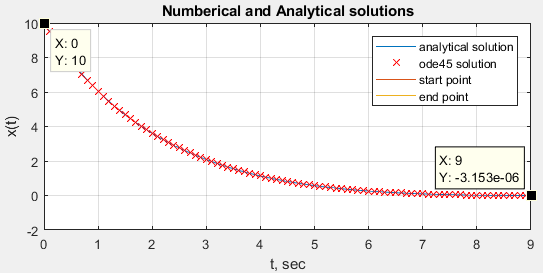


Рисунок 12 – График функции x(t) при таком Um, чтобы x(9) = 0

График входного воздействия u0(t) представлен на рисунке 13.

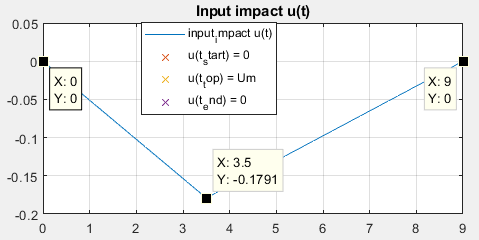


Рисунок 13– График входного воздействия u0(t)

Результат выполнения программы представлен на рисунке 14.

|  |
| --- |
| Um = -0.179125 |

Рисунок 14 – Результат выполнения программы

**Вывод:** решены статическая задача (глобальный максимум в точке x=-2, y = 1) и динамическая (определено значение неизвестного параметра Um=-0.179125) аналитическим и численным методами.