Национальный исследовательский университет «МИЭТ»

Лабораторная работа №1.

Разработка моделей

Москва, 2024

Цель работы

1 Изучения основ построения моделей в MatLab.

2 Изучение приемов построения моделей с графическим пользовательским интерфейсом.

3 Получение практических навыков разработки моделей.

Теоретическая часть

Пример 1. Разработка модели определения скорости пули

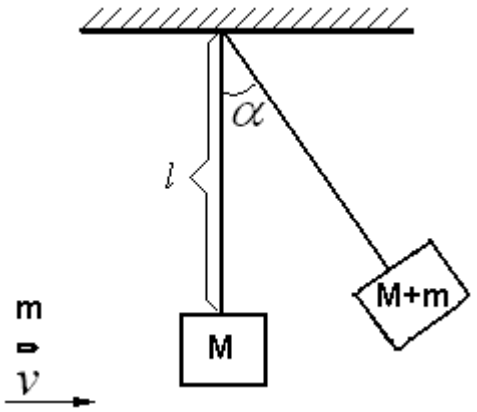


Рисунок 1 – Моделируемая система.

Закон сохранения энергии:

$$\frac{(m+M)\cdot V^2}{2} = (m+M)\cdot g\cdot(l - l\cdot\cos(\alpha)), \qquad (1)$$

где, m – масса пули;

M – масса груза;

l – длина линии подвеса;

g – ускорение свободного падения;

v – скорость пули;

V – скорость системы «пуля-груз»;

α – угол отклонения подвеса от вертикали.

Закон сохранения импульса:

$$m\cdot v = (m+M)\cdot V. \qquad (2)$$

Из уравнения (2) находим начальную скорость системы «пуля-груз»:

$$V = \frac{m\cdot v}{m+M}. \qquad (3)$$

Рассмотрим два варианта: а) при известной скорости пули определить угол отклонения груза; б) по известному углу отклонения груза определить скорость пули.

Подставляя выражение (3) в (1) находим интересующие нас параметры.

а) $$\alpha = \arccos\left(1 - \frac{(m \cdot v)^2}{2 \cdot (m+M)^2 \cdot g \cdot l}\right)$$

б) $$v = \sqrt{\frac{2 \cdot (m+M)^2}{m^2} \cdot g \cdot l \cdot (1 - \cos(\alpha))}$$

Напишем программу, реализующую представленную модель в среде MatLab. Для этого запускаем редактор m-файлов, нажав соответствующую кнопку в панели инструментов.

```
clear; clc;     %очищаем рабочее пространство и экран

M = 10;                 %масса груза, кг
g = 9.8;                %ускорение свободного падения, м/с2
l = 1;                  %длина подвеса груза, м

%модель влияния массы пули на угол отклонения груза
v = 300;                %скорость пули, м/с
m = 0.001:0.001:0.03;
a = acos(1-m.^2*v.^2./(2*(m+M).^2*g*l));
a = rad2deg(a);         %переводим радианы в градусы
subplot(2, 1, 1);
plot(m*1000, a); grid;
xlabel('m, gramm');     %подписываем оси
ylabel('\alpha, \circ');
title('\alpha = f(m).   v = 300 m/c');

%модель скорости пули в зависимости от угла отклонения груза
m = 0.01;               %масса пули, кг
a = 0.1:0.1:14;
v = (2*(m+M)^2*g*l*(1-cos(deg2rad(a)))/(m^2)).^0.5;
subplot(2, 1, 2);
plot(a, v); grid;
xlabel('\alpha, \circ');    %подписываем оси
ylabel('v, m/c');
title('v = f(\alpha).   m = 10 gramm.');
```

Комментарии к программе.

По умолчанию *MatLab* оперирует радианами, поэтому для работы с градусами пользуемся функциями *rad2deg* и *deg2rad*.

Результат моделирования показан на рисунке 2.

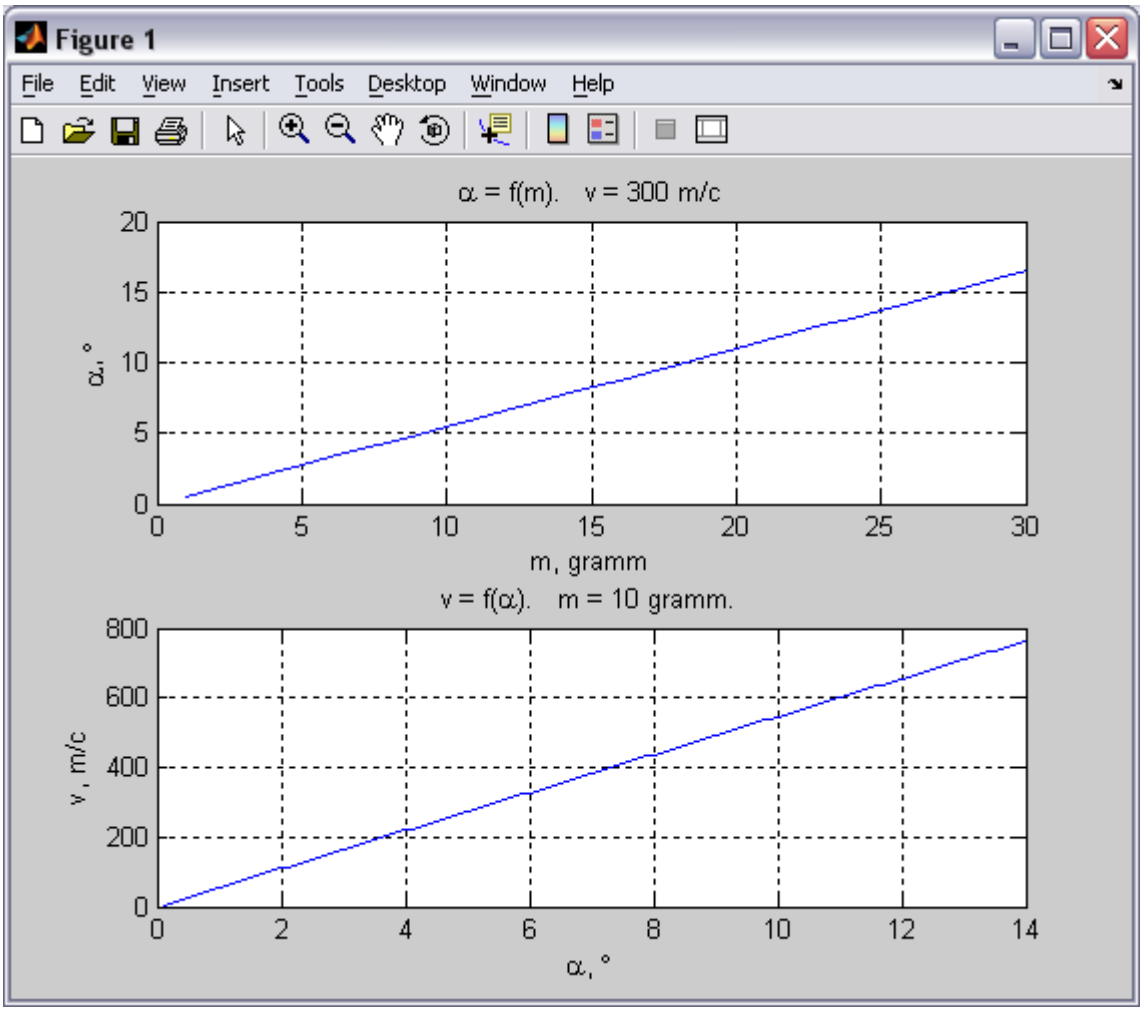


Рисунок 2 – Окно с результатами моделирования.

Разработка моделей с GUI (graphical user interface)

В состав MatLab входит среда GUIDE для создания приложений с графическим интерфейсом пользователя. Работа в этой среде достаточно проста- элементы управления (кнопки, раскрывающиеся списки и т.д.) размещаются при помощи мыши, а затем программируются события, которые возникают при обращении пользователя к данным элементам управления.

Приложение может состоять как из одного основного окна, так и нескольких окон и осуществлять вывод графической и текстовой информации, в основное окно приложения и в отдельные окна. Ряд функций MatLab предназначен для создания стандартных диалоговых окон открытия и сохранения файла, печати, выбора

шрифта, окна для ввода данных и др., которыми можно пользоваться в собственных приложениях.

Что необходимо знать для создания приложений с графическим интерфейсом? Во-первых, как программируются файл-функции с подфункциями, файл-функции с переменным числом входных и выходных аргументов. Во-вторых, требуется иметь представление об иерархической структуре и свойствах графических объектов, уметь обращаться с указателями на них. Разумеется, не должна вызывать затруднение работа с числовыми массивами, строками, структурами, ячейками и массивами строк, структур и ячеек, а также использование конструкций встроенного языка программирования.

Приложение с графическим интерфейсом может быть написано и без применения среды GUIDE. В качестве примера, можно привести bspligui, входящее в состав Spline ToolBox. Желающим разобраться в создании приложений без среды GUIDE можно посоветовать запустить приложение bspligui в режиме отладки, проследить за созданием окна приложения, элементов управления и способе обработки событий (достаточно открыть файл bspligui.m, установить точку останова на первую исполняемую строку и запустить приложение). К этому вопросу мы со временем обратимся.

При создании приложений с графическим интерфейсом пользователя будут полезны следующие разделы справочной системы MatLab:

"MATLAB: Creating Graphical User Interfaces".

"MATLAB: Functions -- Categorical List: Creating Graphical User Interfaces"

"MATLAB: Handle Graphics Property Browser" (справочник свойств графических объектов).

В справочной системе MatLab 7 в разделе "Demo" есть 10-ти минутная демонстрация создания приложения с графическим интерфейсом в среде GUIDE.

Рассмотрим пример создания простой модели для построения графиков некоторых функций.

1. Создание основы.

Перейдите в среду GUIDE, выполнив команду

>> guide

При этом появляется диалоговое окно GUIDE Quick Start (см. рис. 2). У него две вкладки.

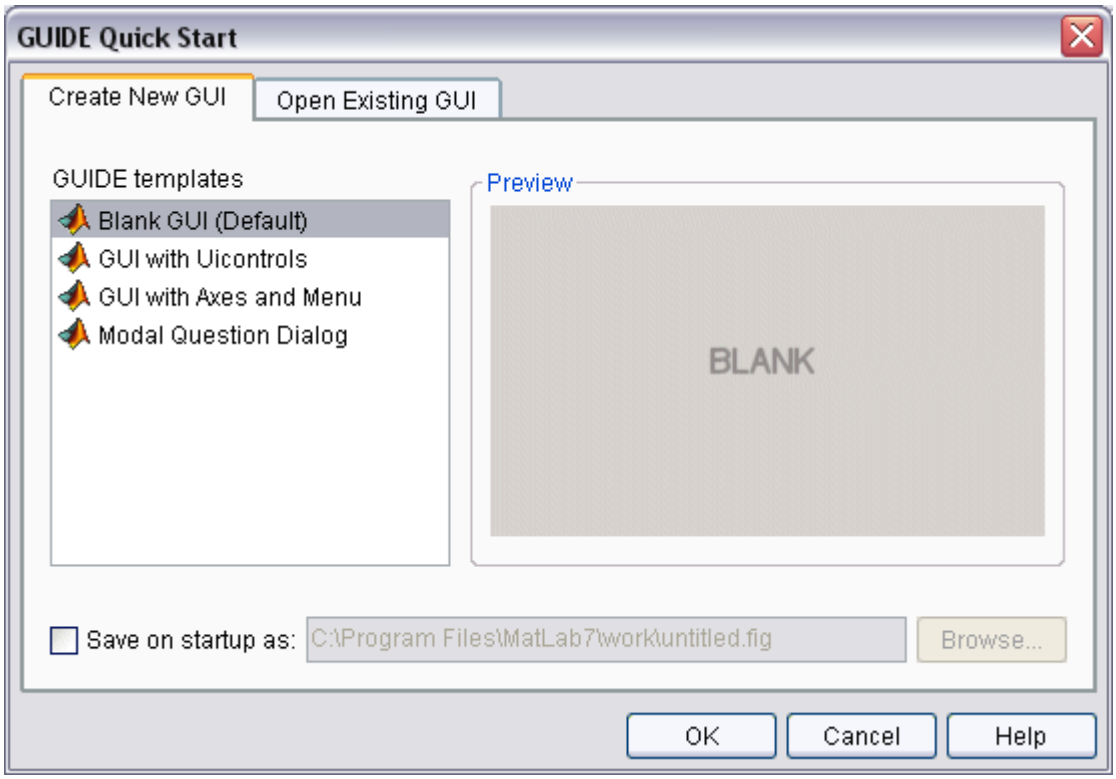


Рисунок 3 – Диалоговое окно GUIDE Quick Start.

Вкладка Create New GUI (создание нового приложения), которая нам сейчас понадобится. На ней можно выбрать четыре заготовки: Blank GUI (пустое окно приложения), GUI with Uicontrols (заготовка с кнопками, переключателями и областями ввода), GUI with Axes and Menu (заготовка с осями, меню, кнопкой и раскрывающимся списком), Modal Question Dialog (заготовка для модального окна).

Вкладка Open Existing GUI (открытие существующего приложения).

На вкладке Create New GUI выбираем строку Blank GUI и нажимаем OK. При этом появляется основное окно среды GUIDE, содержащее заготовку для окна приложения, панель инструментов для добавления элементов интерфейса, управляющую панель и меню (рисунок 4).

Получаем

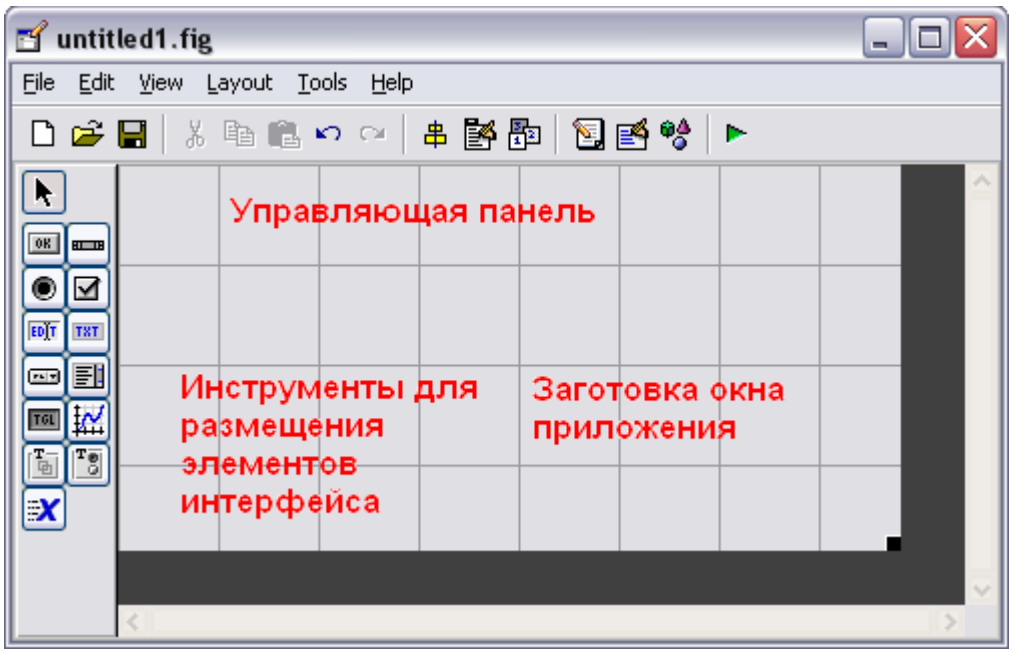


Рисунок 4 – Среда GUIDE с заготовкой для окна приложения.

Размещаем в окне приложения необходимые нам элементы:

- Axes
- Button Group
    - Radio Button
    - Radio Button
    - Radio Button
- Push Button
- Static Text
- Static Text
- Edit Text
- Edit Text

Для этого при помощи мыши выбираем инструмент Push Button (его пиктограмма содержит кнопку OK, а имя появляется на всплывающей подсказке) и щелчком мыши поместите кнопку на заготовку окна приложения

После добавления элемента интерфейса необходимо задать его тег (tag, имя), который будет идентифицировать данный объект среди всех остальных объектов. Тег объекта понадобится для получения и установки его свойств и программирования событий, возникающих при обращении пользователя к элементу управления, например, при нажатии на кнопку.

Для задания тега следует перейти к инспектору свойств. Проще всего это сделать двойным щелчком мыши по добавленному элементу интерфейса. При этом появляется окно инспектора свойств (Property Inspector), в котором отображены свойства элемента управления (объекта Uicontrol). Находим в левом столбце таблицы свойство Tag и в области ввода справа от него изменяем текущее значение на необходимое нам и нажмите <Enter>. Всегда лучше давать объектам содержательные теги!

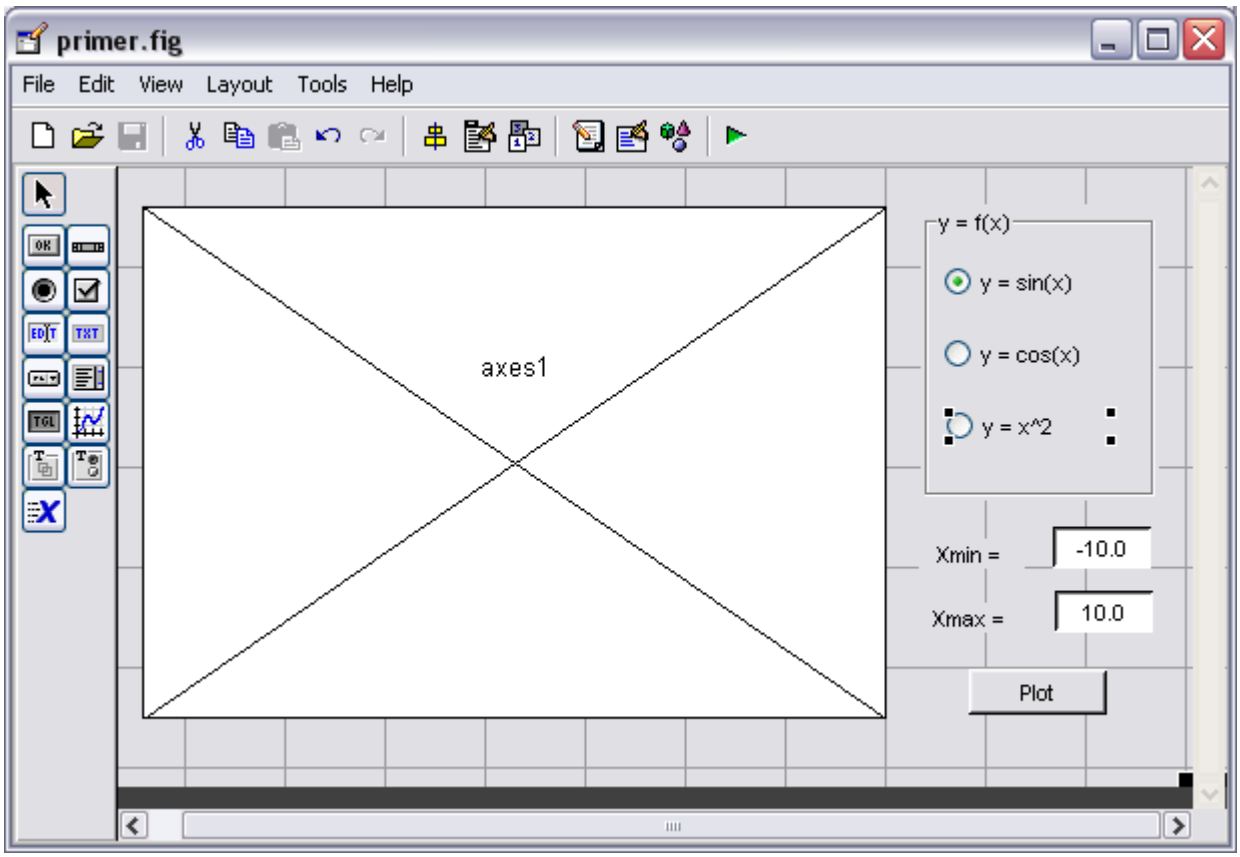
Заметьте, что тег и надпись на кнопке не одно и то же. Сразу же в окне инспектора свойств измените надпись, обратившись к свойству String. Вместо Push Button должно быть Hello

Поменяем теги и надписи на выбранных элементах интерфейса согласно таблице 1.

Объект	Tag	String
Button Group	panel	*) y = f(x)
Radio Button	plot_sin	y = sin(x)
Radio Button	plot_cos	y = cos(x)
Radio Button	plot_x2	y = x^2
Push Button	plot_button	Plot
Static Text	str_Xmin	Xmin =
Static Text	str_Xmax	Xmax =
Edit Text	Xmin	-10.0
Edit Text	Xmax	10.0

*) – Для объекта Button Group свойство String отсутствует. Поэтому изменяем свойство Title.

В итоге получаем окно представленное на рисунке



Сохраним созданную заготовку. Для этого в меню File среды GUIDE выбираем пункт Save As…, появляется диалоговое окно сохранения файла, в котором выберите папку или создайте новую и задайте имя файла primer (автоматически добавится расширение fig). Обратите внимание, что после сохранения приложения в редакторе М-файлов открылся файл primer.m. По умолчанию, приложение содержится в двух файлах: с расширением fig (графическое окно с размещенными на нем элементами управления) и с расширением m (файл-функция primer с подфункциями, которые обрабатывают различные события, возникающие в ходе взаимодействия приложения с пользователем).

Приступим к программированию события Callback кнопки plot_button. Для этого перейдем к заготовке окна приложения и в контекстном меню кнопки выберем в пункте View Callbacks подпункт Callback. При этом происходит переход в редактор М-файлов к подфункции обработки события plot_button_Callback, заголовок которой и комментарии генерируется автоматически:

```
% --- Executes on button press in plot_button.
function plot_button_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to plot_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

После заголовка требуется разместить те операторы, которые будут выполняться при нажатии на кнопку Plot.

Имя файл-функции состоит из тега объекта (plot_button), событие Callback которого будет обрабатываться и названия события Callback (есть и другие события). Смысл ее входных аргументов следующий.

Аргумент hObject содержит указатель на кнопку plot_button, т.е. объект Uicontrol с тегом plot_button (он нам сейчас не понадобится).

Аргумент eventdata зарезервирован для использования в следующих версиях MatLab.

Аргумент handles является структурой с указателями на все объекты приложения. Поля структуры handles являются тегами этих объектов. Так handles.plot_button содержит указатель на кнопку plot_button, handles.figure1 - указатель на окно приложения и т.д.

После заголовка подфункции plot_button_Callback разместим операторы, которые будут строить график, выбранной пользователем функции.

Для работы понадобятся две функции:

get – получение свойств объекта.

set – установка требуемых значений свойств объекта.

В итоге функция будет выглядеть следующим образом:

```
xmin = get(handles.Xmin, 'String');        %считываем минимальное значение x
xmin = str2num(xmin);                      %переводим его в число
xmax = get(handles.Xmax, 'String');        %считываем максимальное значение x
xmax = str2num(xmax);                      %переводим его в число
x = xmin : 0.1 : xmax;                     %задаем вектор значений по оси X
grafik = get(handles.panel, 'SelectedObject');   %какую функцию выбрали
switch (grafik)                                  %строим график
   case {handles.plot_sin}                       %соответствующей функции
      plot(x, sin(x)); grid;
```

```
    case {handles.plot_cos}
        plot(x, cos(x)); grid;
    case {handles.plot_x2}
        plot(x, x.^2); grid;
end
```

Пользователь выбирает исследуемую функцию на панели Button Group (тег - panel). У этого объекта есть свойство SelectObject в котором храниться тэг выбранного объекта. С помощью команды

```
grafik = get(handles.panel, 'SelectedObject');
```

происходит считывание этого тега в переменную grafik. После чего происходит ее сравнение с возможными вариантами.

Теперь приложение hello можно запустить, воспользовавшись кнопкой Run на панели управления среды GUIDE. Перед запуском может появится окно, в котором говорится о том, что папка с файлами приложения не является текущей. В этом окне можно либо сделать ее текущей (переключатель Change MATLAB current directory), либо добавить папку в начало пути поиска MATLAB (переключатель Add directory to the top of the MATLAB path), либо в конец пути поиска (переключатель Add directory to the bottom of the MATLAB path). Установите верхний переключатель (как по умолчанию) и нажмите кнопку OK.

Работающее приложение показано на рисунке

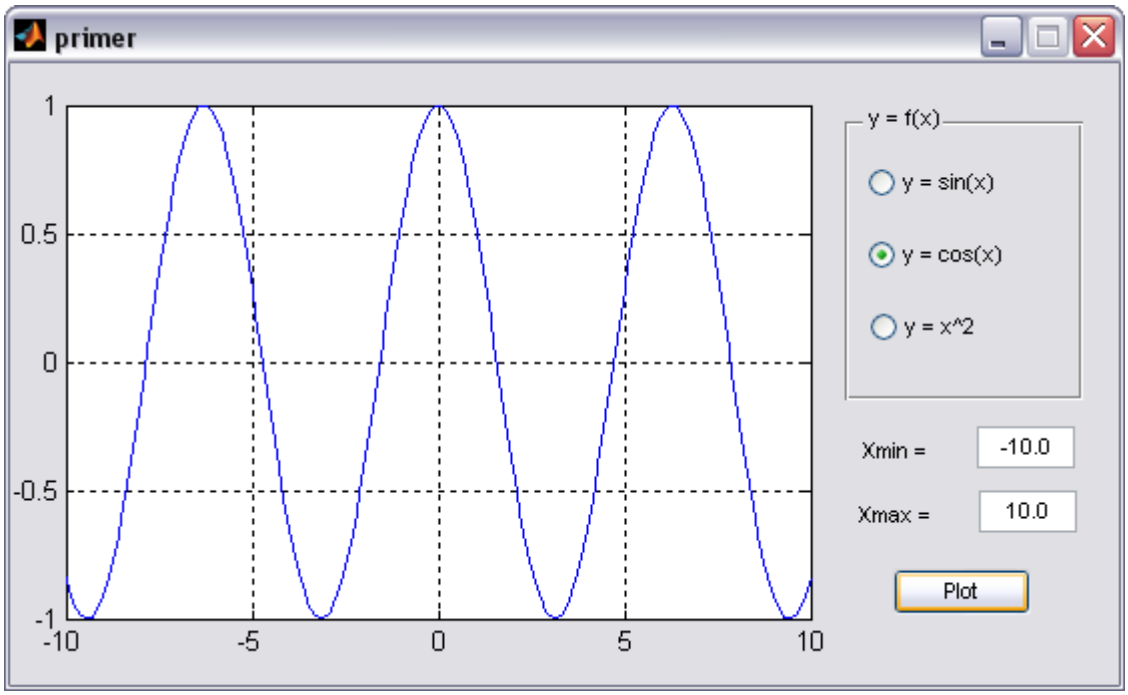


Рисунок 5 – Окно разработанного приложения.

Пример 2. Построение модели сверления слоя металла лазером.

Если энергия лазера полностью идет на испарение столбика металла массы $LS\rho$ (S – облучаемая площадь, L – высота столбика, ρ – плотность вещества), то закон сохранения энергии выражается равенством

$$E_0 = Wt_k = hLS\rho,$$

где h - энергия, требуемая для испарения единицы массы. Величина h имеет составную структуру: $h = (T_{пл} - T)h_1 + h_2 + h_3$ поскольку материал необходимо последовательно нагреть до температуры плавления $T_{пл}$, а затем расплавить и превратить в пар (T – исходная температура, h_1 – удельная теплоемкость, h_2 и h_3 – соответственно удельная теплота плавления и парообразования).

Изменение глубины выемки $l(t)$ со временем определяется из детального баланса энергии в промежутке времени от t до $t + dt$. На испаренную за это время массу

$$[l(t + dt) - l(t)]S\rho = dlS\rho$$

тратится энергия $dlhS\rho$, равная энергии Wdt, сообщаемой веществу лазером:

$$dlhS\rho = Wdt$$

откуда получается дифференциальное уравнение

$$\frac{dl}{dt} = \frac{W}{hS\rho}$$

Его интегрирование (с учетом того, что начальная глубина выемки равна нулю) дает

$$l(t) = \frac{W}{hS\rho}t = \frac{E(t)}{hS\rho},$$

где $E(t)$ – вся энергия, выделенная лазером к моменту времени t. Следовательно глубина выемки пропорциональная затраченной энергии.

В действительности процесс сверления гораздо сложнее рассмотренной схемы – энергия тратится на нагрев вещества, на удаление паров из выемки, которая может иметь неправильную форму, и т.д. Однако для первоначальной оценки времени сверления отверстия модель вполне подходит.

Разработка модели в MatLab.

Практическая часть

Задание 1

Построить модель, описанную в примере 1. Получить трехмерные графики зависимостей изменения угла отклонения груза, от различных параметров (параметр определяется по номеру варианта по таблице, приведенной ниже).

№Варианта	Параметр
1, 2, 3, 4, 5	Масса пули.
6, 7, 8, 9, 10	Масса груза.
11, 12, 13, 14, 15	Длина подвеса.
16, 17, 18, 19, 20	Начальная скорость пули.
21, 22, 23, 24, 25	Ускорение свободного падения.

Задание 2

Разработайте программу с GUI, которая позволяет пользователю выбрать одну из тригонометрических функций и построить ее график.

Задание 3

Разработайте модель в MatLab с применением GUI. Пример возможного внешнего вида модели представлен на рисунке 6.

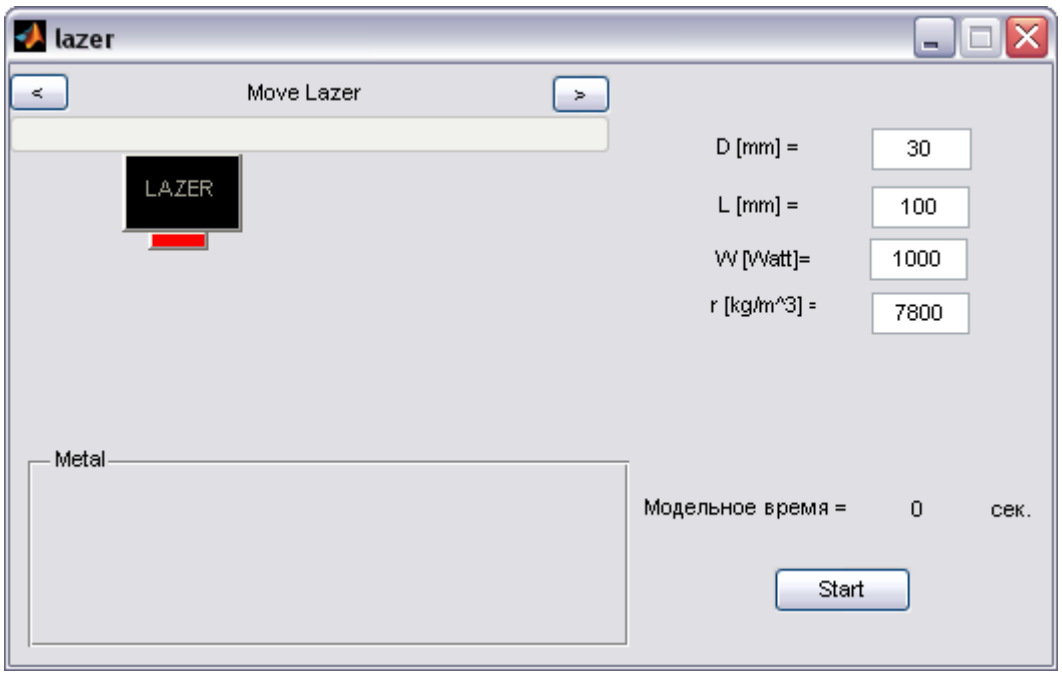


Рисунок 6 – Вид разработанной модели.

Исходными данными для модели являются:

– диаметр отверстия (мм);

– толщина металла (мм);

– мощность лазера (Вт);

– плотность металла (кг/м^3).