В. С. ШАРАЙ, В. В. ДАВЫДОВСКАЯ
УО МГПУ им. И. П. Шамякина (г. Мозырь, Беларусь)

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ СОВРЕМЕННЫХ ИНТЕГРИРОВАННЫХ ПАКЕТОВ ДЛЯ МОДЕЛИРОВАНИЯ ФИЗИЧЕСКИХ СИСТЕМ

В общем случае компьютерное моделирование физических процессов и систем является весьма сложной задачей. Помимо умений в области программирования необходимы глубокие знания в области физики и математики (методов численного решения). Для моделирования простейшей физической задачи необходимо иметь представления о дифференциальных уравнениях и способах их численного решения, что зачастую является проблематичным. Пакет прикладных программ MATLAB упрощает задачу программирования своими встроенными функциями для решения целого ряда задач определенного типа. При моделировании в этой среде нет необходимости в идеале владеть численными методами решения дифференциальных уравнений, что не только упрощает задачу, но и значительно уменьшает количество строк в коде программы. Одной из важных частей моделирования является визуализация решения (графики, таблицы, диаграммы). Построение графических изображений является трудно реализуемой задачей на таких языках программирования, как C, C++, Delphi и др., чего нельзя сказать про MATLAB, где функция построения графика является простой и удобной в использовании.

Таким образом, преимущества MATLAB при моделировании физических процессов и систем перед другими средами программирования очевидны. Рассмотрим это на примере задачи колебания математического маятника. Математическим маятником называется малое тело (материальная точка) массы m, подвешенное на нити длиной l. Во многих источниках приводится уравнение для описания процесса колебания математического маятника (см., напр. [1]):

$$\frac{d^2\varphi}{dt^2} + \omega^2 \cdot sin\varphi = 0, \quad (1)$$

где φ – угол отклонения маятника, $\omega^2 = g/l$, g – ускорение свободного падения, l – длина нити маятника.

Рассмотрим случай малых колебаний. Пусть в начальный момент маятник отклонён от вертикали на угол φ_0 и отпущен без начальной скорости. Тогда начальные условия будут:

$$t=0,\ \varphi=\varphi_0,\ \dot{\varphi}_0=0. \quad (2)$$

Для малых колебаний (φ<10°) будем считать $sin\varphi \approx \varphi$. Тогда уравнение (1) примет следующий вид:

$$\frac{d^2\varphi}{dt^2} + \omega^2 \cdot \varphi = 0. \quad (3)$$

Уравнение (3) есть дифференциальное уравнение простого гармонического колебания. Общее решение этого уравнения имеет вид:

$$\varphi = A\cos\omega t + B\sin\omega t, \quad (4)$$

где A и B – постоянные интегрирования [2].

Для решения данной задачи в Delphi необходимо составить явную разностную схему для решения дифференциального уравнения (3). Для этого заменяем вторую производную ее конечно-разностным выражением.

$$\ddot{\varphi} = \frac{\varphi_{i-1} - 2\varphi_i + \varphi_{i+1}}{h^2} \quad (5)$$

где i = 1, 2, 3 ... n.

За нулевой элемент φ_0 берем первоначальное отклонение маятника от положения равновесия. Количество элементов n будем брать исходя из рассматриваемого промежутка времени T и желаемого шага h. При этом примем $\varphi_1 = \varphi_0 - lh$. Таким образом, имеем следующее уравнение:

$$\frac{\varphi_{i-1} - 2\varphi_i + \varphi_{i+1}}{h^2} + \frac{g}{l}\varphi_i = 0.$$

Перепишем это выражение в виде:

$$\varphi_{i+1} = \varphi_i(2 - \frac{g}{l}h^2) - \varphi_{i-1}. \quad (6)$$

Таким образом, получили готовое выражение для поиска угла отклонения маятника в любой момент времени. Программный код вычислительного блока в Delphi выглядит следующим образом:

```
begin
  l:=Strtofloat(Edit1.Text);
  x:=Strtofloat(Edit2.Text);
  Time:= Strtofloat(Edit3.Text);
      n:=500;
      h:=Time/n;
      k:=(9.8/l);
      v:=0;
      alpha:=x/l*pi/180;
u[1]:=x;
u[2]:=u[1]-v*h;
t[1]:=0;
t[2]:=h;
vt[1]:=v;
for i:=2 to n do begin
  t[i+1]:=t[1] + i * h;
  u[i+1]:=(2*u[i] - h*h*(k)*Sin((u[i]/l)*pi/180)-u[i-1]);
end;
 t[n]:=Time;
for i:=1 to n-1 do begin
  vt[i+1]:=(u[i+1]-u[i])/h;
  end;
 vt[n]:=(u[n]-u[n-1])/h;
 for i:=2 to n-1 do begin
  at[i+1]:=(vt[i+1]-vt[i])/h;
  end;
  at[1]:=(vt[3]-vt[2])/h;
  at[2]:=(vt[4]-vt[3])/h;
```

В итоге, запуская готовое приложение, мы можем убедиться в его работоспособности. Результат можно увидеть на рисунке 1.

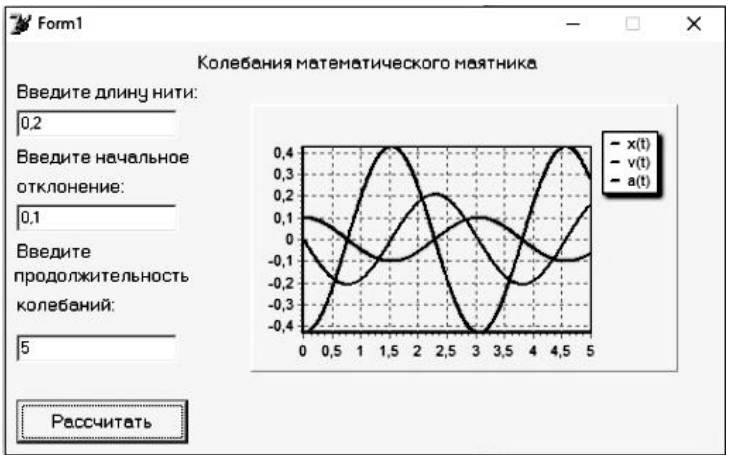

Рисунок 1. – Приложение по моделированию колебаний математического маятника в Delphi

Как видно из приведенного листинга кода, алгоритм решения задачи достаточно сложный для разработчика, если он не знаком с численными методами, а именно с методом конечных разностей. При этом вычисления выглядят громоздкими. Для решения этой же задачи в MATLAB можно воспользоваться встроенной функцией для решения ОДУ – `ode45`. Синтаксис этой функции выглядит следующим образом [3]:

```
[t,y] = ode45(odefun,tspan,y0)
```

Здесь `tspan = [t0 tf]`. Интегрирование системы дифференциальных уравнений `y' = f(t,y)` происходит от начального момента времени `t0` до конечного `tf` с начальными условиями `y0`. Функция `ode45` фактически убирает необходимость численного решения дифференциального уравнения вручную. Параметр `odefun` будет находиться в отдельном m-файле. Его структура будет следующая:

```
function dxdt=fun(t,x)
global l xn g T
dxdt=zeros(2,1);
dxdt(1)=x(2);
dxdt(2)=-(g/l)*sin(x(1)/l*pi/180);
```

В результате будем иметь приложение, изображенное на рисунке 2.

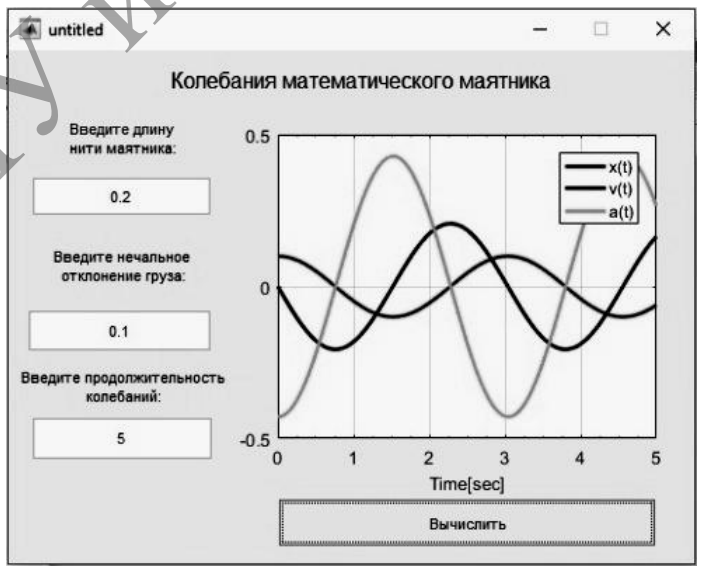

Рисунок 2. – Приложение по моделированию колебаний математического маятника в GUIDE

Листинг Callback-функции для кнопки **Вычислить**:

```
function pushbutton1_Callback(hObject, eventdata, handles)
global l xn g Tp
l=str2double(get(handles.edit1,'string'));
xn=str2double(get(handles.edit2,'string'));
Tp=str2double(get(handles.edit3,'string'));
g=9.8; x0=[xn 0]; N=500;
tspan=linspace(0,Tp,N);
[T,X]=ode45('fun',tspan,x0);
dt=(tspan(end)-tspan(1))/N;
s=length(X(:,1));
f=X(:,2);
A=zeros(s,1);
A=(f(2:s)-f(1:s-1))/dt;
A=[A(1); A];
plot(handles.axes1,T,X(:,1),'r','LineWidth',2)
hold on
plot(handles.axes1,T,X(:,2),'b','LineWidth',2)
plot(handles.axes1,T,A,'g','LineWidth',2)
set(handles.axes1,'XMinorTick','on')
grid(handles.axes1,'on')
box(handles.axes1,'on')
xlabel(handles.axes1,'Time[sec]','FontSize',10)
legend(handles.axes1,'x(t)','v(t)','a(t)', 4)
```

Следует отметить, что при визуально одинаковых полученных результатах (рисунки 1 и 2), для решения этой задачи в Delphi, составлялась явная разностная схема для решения дифференциального уравнения, что также требует знаний в области численных разностных методов для решения дифференциальных уравнений [1]. В MATLAB есть встроенные функции для решения дифференциальных уравнений и их систем.

Сравнивая вычислительные блоки обеих программ, можно сделать выводы о простоте моделирования физических и математических задач в программе MATLAB, используя встроенные функции и пакеты для вычислений. Для проведения аналогичных действий в решении задачи в Delphi приходится вручную создавать алгоритм численного решения дифференциального уравнения, что является достаточно сложной задачей.

ЛИТЕРАТУРА

1. Бухгольц, Н. Н. Основной курс теоретической механики / Н. Н. Бухгольц. – М. : Наука, 1969. – 336 с.
2. Лазарев, Ю. Ф. Моделирование процессов и систем в MATLAB : учеб. курс / Ю. Ф. Лазарев. – СПб. : Питер, 2005. – 511 с.
3. Список функций PDE Toolbox [Электронный ресурс]. Режим доступа: http://MATLAB.exponenta.ru/pde/book3/1/pdetool.php. – Дата доступа: 05.02.2020.