Использование системы MATLAB для решения математических задач.

Д. К. Разманова
Руководитель В. П. Поплёвко

MATLAB - это пакет прикладных программ для решения задач в областях, требующих особой надежности при обработке данных. Язык MATLAB - это инструмент, обеспечивающий взаимодействие оператора для анализа, сбора и представления данных.

MATLAB помогает решать многие задачи. Например, находить интегралы. Для вычисления интегралов используются следующие наиболее известные методы: 1)метод трапеций, его реализуют функции cumtrapz() и trapz(). Функцию cumtrapz() используют при работе с табличными данными или векторами. Результатом является n-интегралов, образующих вектор. Данная ф-я накапливает сумму значений - это удобно для графического отображения. Пример: y = e^x + 2

```
>> x = 0:0.01:3; % диапазон интегрирования
>> y = exp(x) + 2; % значения подынтегральный функции
>> F = cumtrapz(x,y); % постепенный расчет интеграла
>> plot(x, y, 'b', x, F, 'r')  % построение функции
```

Функция trapz() отличается от cumtrapz() тем, что вычисляет сумму всех значений. Пример: y(x) = x*e^x + ln(x) + 1

```
>> x=1:10; % диапазон интегрирования
>> trapz(x.*exp(x) + log(x) +1); % x. - обозначение матрицы
ans = 2.1806e+05
```

2) метод Симпсона. Результат обладает более высокой точностью, чем в методе трапеций. Функция integral(). Пример: f(x) = ln(x)

```
>> fun = @(x)log(x);
>>  integral(fun, 0, pi)
ans = 0.4547
```

3) символьное интегрирование определенных и неопределенных интегралов. Функция int().

```
>> syms x % создание символьной скалярной переменной
>> int((15*x^2-4*x-81)/((x-3)*(x+4)*(x-1)))
ans = 7*log(x - 1) + 3*log(x - 3) + 5*log(x + 4)
```

Рассмотрим функции MATLAB для решения дифференциальных уравнений. Для решения ОДУ используют функцию dsolve. Формат ее вызова: dsolve (equation, condition), где equation – строка, задающая уравнение, с независимой переменной, а condition - одно или несколько условий (необязательный параметр). Примеры: 1) уравнение Бернулли x^2*y' = y*(x + y)

```
>> syms y(x) % создание символьной скалярной функции
>> dsolve(diff(y, x) == (y*(x+y))/x^2)
ans = x/(C1 - log(x))
```

0

2) y'' = a^2*y, где y(0) = b, y'(0) = 1

>> syms y(x), a, b % создадим символьную скалярную функцию, переменные

>> ans(x) = dsolve(diff(y, x, 2) == a^2 * y, [y(0) == b, diff(y, x) == 1]) % продифференцируем

ans(x) = (exp(a*x)*(a*b + 1))/(2*a) + (exp(-a*x)*(a*b - 1))/(2*a)

При решении уравнения и условия могут содержать неопределенные переменные. В таком случае эти переменные являются символьными и входят в общее решение. Если же данное уравнение содержит неопределенную функцию, то в решении появляются неопределенные интегралы.

Решая ОДУ, функция dsolve пытается найти решение в явном виде, если ей это не удается, то выводится сообщение, и возвращается пустой символьный объект. Но, если указать при вызове опцию ‘Implicit’, ‘true’, то можно найти решение в неявном виде. Пример: (y + e^y)*y' = x - e^(-x)

>> dsolve(diff(y) == (x - exp(-x)) / (y(x) + exp(y(x))))

Warning: Unable to find symbolic solution.

> In dsolve>assignOutputs (line 235)

In dsolve (line 221) ans =[empty sym]

Решение в неявной форме:

>> dsolve(diff(y) == (x-exp(-x))/(y(x)+exp(y(x))), 'Implicit', true)

ans = exp(y(x)) + y(x)^2/2 == C1 + exp(-x) + x^2/2

В MATLAB существуют солверы для решения ОДУ, вычисляющие системы уравнений вида y' = f(x, y). Для численного решения задачи Коши обыкновенных дифференциальных уравнений или систем таких уравнений используются следующие солверы (solver): ode23, ode45, odell3, odel5s, ode23s, ode23t и др. Формат вызова:

[X, Y] = odeXY(odefun, ,[t_0, t_end], y_0, options), где odeXY – имя одной из функций - солверов, odefun – дескриптор функции, [t_0, t_end] - диапазон изменения независимого переменного, y_0 - вектор начальных условий, options - параметры вычислительного процесса. Для корректировки решения можно управлять следующими параметрами солверов: точностью вычислений, шагом интегрирования, выходными данными, событиями, и др. В результате вычисления солвер возвращает вектор X с координатами значений переменной, в которых найдено решение, и матрицу решений Y, где каждый столбец - это значение функции в независимой переменной из вектора.

Выбор решателя зависит от жесткости задачи (трудности при вычислении) и происходит подбором, а затем оценкой полученного результата. Первым из списка будет ode45, который вычисляет большинство ОДУ. При решении дифференциальных уравнений и систем с начальными условиями MATLAB следует правильно выбирать солвер, чтобы избежать получения неточного результата. Примеры: 1) создадим функцию в отдельном файле, а затем вызовем ее:

```
>> [X, Y] = ode23(@odefn, [-1.5, 1.5], 0.01);
>> plot(X, Y), grid on % Построение графика с разметкой
```

2) функция в виде аргумента при вызове: y = -5*x^3 - 3/x

```
>> [X, Y] = ode45(@(x, y) -5*x^3-3/x, [-2 2], 0);
>> plot(X, Y), grid on
```

В данном докладе были приведены основные сведения о функциях MATLAB для нахождения интегралов, решения дифференциальных уравнений и задачи Коши. Для обыкновенных дифференциальных уравнений и их систем существует множество других функций, таких как, солвер bvp4c - для решения краевых задач ОДУ, пакет расширения PDE TOOLBOX - для численного решения, пакеты расширений, предназначенные для исследования математических моделей, сводящихся к решению систем дифференциальных уравнений. Одним из таких пакетов является пакет SIMULINK. Также в MATLAB встроен язык MAPLE.