

ВСТРОЕННЫЕ ФУНКЦИИ СИСТЕМЫ MATLAB ДЛЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

В системе MATLAB есть большое количество пакетов расширения (TOOLBOX), которые многократно увеличивают эффективность использования системы и позволяют решать задачи из различных научных областей [1]. Одним из таких пакетов является PDE TOOLBOX, предназначенный для решения дифференциальных уравнений в частных производных (ДУЧП) и их систем [2].

Приближенным решением граничной задачи является вектор значений в узлах сетки. Его создают специальные функции пакета PDE Toolbox называемые солверами (solver). Перед ее вызовом должна быть создана матрица или файл-функция граничных условий b , заданы триангуляция в массивах p , e , t и заданы коэффициенты уравнения (разумеется, переменные могут называться по-другому). Если решается нестационарная задача, то необходимо создать функции, определяющие решение в начальный момент времени и вектор со значениями времени, в которые следует найти решение. Затем эти данные передаются функциям солверам. Они находят приближенное решение задачи в узлах сетки. Солверы реализованы в специальных файл-функциях – для каждого типа задачи свой солвер.

В следующей таблице перечислены основные солверы [3].

Таблица 1 – Основные специальные функции пакета PDE Toolbox

Имя функции	Описание типа решаемой задачи
asempde	Эллиптические уравнения и системы
parabolic	Параболические уравнения и системы
hyperbolic	Гиперболические уравнения и системы
pdenonlin	Нелинейные стационарные уравнения
adaptmesh	Адаптивная генерация сетки и решение эллиптических уравнений и систем с заданной точностью
pdeeig	Решение эллиптических задач на собственные значения

Интерфейс солвера **asempde** является наиболее универсальным. Она решает уравнение (или систему) вида – $\text{div}(c*\text{grad}(u))+a*u=f$. Первым его входным аргументом является матрица или строка с именем файл – функции граничных условий. Далее указываются матрицы с информацией о триангуляции и коэффициенты уравнения c , a , f . Ее вызовы с одним выходным аргументом

$u = \text{asempde}(b, p, e, t, c, a, f);$ % b – матрица

u = assempde('bouncond', p, e, t, c, a, f); % bouncond - функция приводят к записи в вектор **u** значений решения в узлах сетки, координаты которых хранятся в **p**.

Входные аргументы солверов **parabolic** и **hyperbolic** такие же, как у **assempde**. Кроме того, задается вектор значений решения **u0** в начальный момент времени (в случае гиперболического уравнения еще и вектор **ut0** с начальным значением производной решения по времени), вектор **tlist** моментов времени, в которые требуется найти решение, и коэффициент **d** уравнения при второй производной по времени. Эти солверы возвращают матрицу **u** с решением, каждый столбец которой есть вектор со значениями приближенного решения в узлах сетки в соответствующие моменты времени, указанные в **tlist**:

u = parabolic(u0, tlist, b, p, e, t, c, a, f, d)

u = hiperbolic(u0, ut0, tlist, b, p, e, t, c, a, f, d)

Визуализация результатов.

Функции **pdegplot** и **pdemesh** предназначены для графического отображения геометрии области и сетки, их использование описано выше. Основной функцией для визуализации решения является **pdeplot**, которая позволяет управлять видом получаемых графиков при помощи ряда дополнительных параметров. Функции **pdecont** и **pdesurf**, строящие линии уровня решения или трехмерный график, реализуют частные случаи обращения к **pdeplot**. Входными аргументами **pdecont** и **pdesurf** являются матрица **p** с координатами узлов, матрица **t** соответствия глобальной и локальной нумераций и вектор **u** со значениями решения в узлах. Четвертым дополнительным аргументом **pdecont** может являться вектор значений, которые требуется отобразить линиями уровня, или их число.

Первые три аргумента функции **pdeplot** являются массивами **p**, **e** и **t** с информацией о триангуляции. Затем парами задаются имя параметра (свойства) и его значение.

Для иллюстрации использования описанных выше функций решим несколько задач в полукруге. Центр полукруга находится в начале координат, радиус единица, выбрана верхняя часть круга. Вначале создадим функцию декомпозиционной геометрии области, при этом учтем, что область должна задаваться как минимум двумя граничными кривыми.

Проверяем, что граница области создана правильно

pdegplot('halfcirclegeom'), axis equal

Строим и рисуем сетку

[p, e, t] = initmesh('halfcirclegeom');

pdemesh(p,e,t); axis equal;

Сгустим и нарисуем сетку (правый рисунок)

```
[p,e,t]=refinemesh('halfcirclegeom',p,e,t);
[p,e,t]=refinemesh('halfcirclegeom',p,e,t);
pdemesh(p,e,t); axis equal;
```

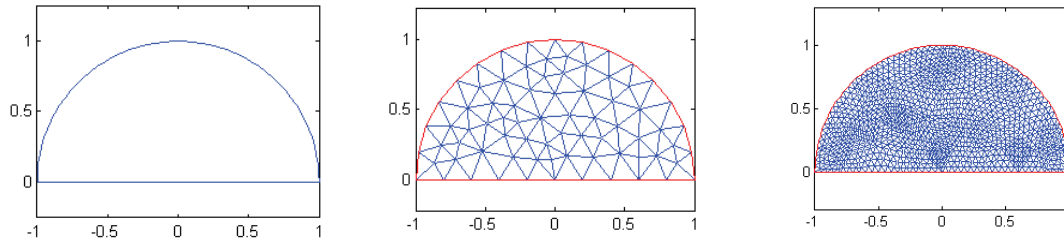


Рисунок 1 – Область для решения краевых задач в полукруге

Синтаксис функции аналогичен функции ASKER

Решим уравнение Пуассона $\Delta u = -1$ с нулевыми граничными условиями. Для граничных условий Дирихле $h*u = g$ на обоих участках границы (на диаметре и полуокружности) имеем $h=1$, $g=0$. Матрицу граничных условий можно построить командой

```
b = [1 1; 1 1; 1 1; 1 1; 1 1; 1 1; 48 48; 48 48; 49 49; 48 48]
```

Здесь числа 48 и 49 являются кодами символов 0 и 1, представляющих выражения для g и h .

Эллиптическое уравнение $-\text{div}(c*\text{grad}(u)) + a*u = f$ принимает требуемую форму при $a=0$, $c=1$, $f=1$. Задаем коэффициенты и правую часть уравнения $-\Delta u = f$.

Для решения эллиптических уравнений используется функция `assemprde`. Ей следует передать аргументы – матрицу граничных условий b , массивы p , e , t , содержащие информацию о триангуляции, и коэффициенты уравнения c , a , f . Построим график поверхности решения $u(x,y)$ с помощью `pdesurf(p,t,u)`.

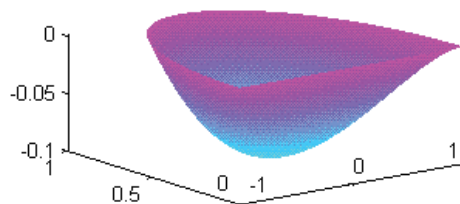


Рисунок 2 – График поверхности решения уравнения Пуассона.

ЛИТЕРАТУРА

1. Дьяконов В. П. MATLAB. Полный самоучитель. – М.: ДМК Пресс, 2012. – 768 с
2. Гилат Амос. MATLAB. Теория и практика. 5-е изд. / Пер. с англ. Смоленцев Н. К. - Москва : ДМК Пресс, 2016. - 416 с.
3. Джон Г. Мэтьюз, Куртис Д. Финк. Численные методы. Использование MATLAB. М.: Издательский дом «Вильямс», 2001. – 720 с.