МАТЕМАТИКА И МАТЕМАТИЧЕСКО ОБРАЗОВАНИЕ, 2010
MATHEMATICS AND EDUCATION IN MATHEMATICS, 2010
Proceedings of the Thirty Ninth Spring Conference of the Union of Bulgarian Mathematicians
Albena, April 6–10, 2010

ИЗПОЛЗВАНЕ НА СИСТЕМАТА MATLAB В ТЕХНИЧЕСКИТЕ УНИВЕРСИТЕТИ*

Михаил Константинов, Владимир Тодоров, Галина Пелова, Юлиана Бонева

Разгледани са някои аспекти на обучението по математика с помощта на програмната система MATLAB[1] в техническите университети. Специално внимание е отделено на разделите аналитична геометрия и диференциални уравнения

Уводни бележки. MATLAB (от MATrix LABoratory) е универсална програмна система за числени и символни пресмятания [3, 1, 2] (справка в Google показва над 16 млн споменавания на системата. Подобни са системите Maple (над 73 млн споменавания) и MATHEMATICA (над 4 млн споменавания). Актуализиран към 2009 г. списък от 1 200 заглавия на книги за MATLAB и нейните приложения (включително от български автори) може да се намери на адрес
`http://www.mathworks.com/support/books/index.jsp`.

Системата MATLAB се отличава с изключително приятелски интерфейс, което я прави удобна за обучение по различни математически и инженерни дисциплини в техническите университети. Ядро на MATLAB са надеждните числени алгоритми за решаване на основните задачи на матричната алгебра.

MATLAB може да се използва и в четирите традиционни за техническите и икономическите университети математически дисциплини: "Линейна алгебра и аналитична геометрия" (ЛААГ), "Математически анализ 1", "Математически анализ 2" и "Приложна математика".

В среда на MATLAB могат да се осъществяват сложни операции с вектори и матрици и така напълно се покрива частта "линейна алгебра " от курса по ЛААГ. Същевременно това дава възможност да се реализират програмно алгоритми за решаване на задачи и от аналитичната геометрия. Целта тук е да се улеснят пресмятанията и да се подпомогне изучаването на основните понятия и факти от преподавания курс. Нещо повече, работата с подобна система помага за по-доброто усвояване на тези понятия, тъй като упражненията с компютър за днешните студенти са лесно и приятно занимание. За съжаление, не такъв е случаят с някои учители и университетски преподаватели.

---

*Тази работа е подпомогната по дог. 102/2009 с НИПС на УАСГ

[1]MATLAB© е търговска марка на MathWorks, Inc.

2000 Mathematics Subject Classification: Primary 97D40
Key words: MATLAB, аналитична геометрия, диференциални уравнения

Други естествени приложения на MATLAB са аналитичното и численото решаване на начални и гранични задачи за обикновени диференциални уравнения (ОДУ), както и численото решаване на начални задачи за частни параболични и елиптични диференциални уравнения.

Аналитична геометрия с MATLAB. Дали и как трябва да се преподава аналитична геометрия в техническите университети е тема, която не разглеждаме. Както е известно, упражненията по аналитична геометрия се основават на работата с вектори. Но студентите в първи курс имат бегли знания, или въобще не са запознати с темата "Вектори" от гимназиалния курс. Това създава известни затруднения, особено при решаването на задачи в пространството, като например намирането на ос на две кръстосани прави. Използването на системата MATLAB за решаване на такива задачи дава възможност примерите, които се дават на студентите, да не са твърде "учебни", т.е. вече не е нужно данните и резултатите да са целочислени, за да може да се смята "на ръка". На свой ред използването на графичния режим дава нагледна представа и помага за по-доброто преподаване на материала, свързан с кривите и повърхнините от втора степен.

И така, идеята да използваме системата MATLAB е добра, но как точно да я реализираме? Първо, трябва известна подготовка за преструктуриране на преподавания материал. Необходимо е включване на упражнения за първоначално запознаване със системата и работа в директния режим за въвеждане на данни от клавиатурата. Следва запознаване с операциите с вектори и матрици, включително и с нестандартните операции, включени в MATLAB.

Още в началото разглеждаме скаларното и векторното произведения. Скаларното произведение на два вектора v_1 и v_2 с еднаква дължина се пресмята с функцията (командата) `dot`, а векторното произведение на два вектора в тримерното пространство с функцията `cross`. Въвеждането на векторите и пресмятанията се извършват както следва

```
>> v1 = [1 2 3]; v2 = [4 5 6]; a = dot(v1,v2)
a =
    32
```

Тук с дидактична цел пресмятаме скаларното произведение и с командата `a = v1*v2`, което дава същия резултат.

По-нататък имаме

```
>> cross(v1,v2)
ans =
    -3     6    -3
```

Тук се появява променливата `ans`, с която се означава текущият резултат, ако преди това не сме му дали наименование.

Трябва да обърнем внимание, че в MATLAB са предвидени редица полезни операции, позволяващи да се преобразува даден масив (вектор или матрица) в друг масив, имащ същия размер и тип. Тези операции не са стандартни и не се изучават в курса по математика. Към тях се отнасят в частност всички операции, осъществявани с помощта на вградените математически функции на един аргумент. Например, ако x е даден вектор, то командата `y = sin(x)` води до получаването на нов вектор y, чиито елементи са равни на синусите от елементите на x.

Освен това са предвидени някои операции на поелементно преобразуване на масиви, наречени "точкови" операции (dot operations). Тези операции позволяват лесно да се пресмятат сложни числови математически функции, а след това да се построят графиките им, без да се използва оператор за цикъл, т.е. в режим калкулатор. Това е една добра подготовка на студентите за упражненията по математически анализ – построяване на графика на дадена функция.

Следваща стъпка в обучението е запознаване с т. нар. *М–файлове* с цел създаване на програми на езика на MATLAB. Има два вида М–файлове: *файл сценарий (управляваща програма)* и *файл функция (процедура)*. Това дава възможност за реализиране на многостъпкови алгоритми. За всяка стъпка се написва М–файл функция и сглобяването се извършва в управляващата програма.

Например, задачата за намиране на симетричната точка p' на дадена точка p относно дадена права $\ell = \{a + nt : t \in \mathbb{R}\}$ в пространството $\mathbb{R}^3$, където $a, n \in \mathbb{R}^3$ са зададени вектори ($n \neq 0$), може да се реши по следния начин (скаларният параметър t се интерпретира като *време*, а x – като *текуща точка*, движеща се по правата):

1. Създаваме М–файл функция за намиране на общото уравнение на равнината λ през точка p с нормален вектор n; имаме $\lambda = \{x \in \mathbb{R}^3 : n^\top(x - p) = 0\}$ (в означенията на MATLAB това е уравнението `dot(n,x-p) = 0`).
2. Създаваме М–файл функция за намиране на пресечната точка q на правата ℓ и равнината λ; това се случва в момента $t_q = n^\top(p - a)/n^\top n$, или `tq = dot(n,p-a)/dot(n,n)`.
3. Създаваме М–файл функция за намиране на точката p', която е край на отсечка, за която се знае другият край p и средата q; това впрочем е положението $p' = p + 2t_q$, в което изобразяващата точка попада в момента $2t_q$.
4. Създаваме М–файл сценарий, в които се въвеждат входните данни и се викат създадените вече функции.

Впрочем, така се решава задачата и в пространството $\mathbb{R}^m$ за всяко $m \geq 2$.

На адрес `http:/www.uacg.bg/books/matlab-ag` е дадена библиотечка от М–файлове за решаване на задачи на аналитичната геометрия в среда на MATLAB.

Решаване на диференциални уравнения с MATLAB . Ефективно средство за описание и изучаване на явленията във всички области на естествознанието е математическото моделиране с помощта на диференциални уравнения. Изобщо, нещата, които студентите в техническите университети трябва да "отнесат" със себе си от курса по математика, са минимум три: *матрична алгебра*; *производни и интеграли*; *диференциални уравнения*. Тези три теми първо трябва да се поднесат теоретично, а след това да се покаже как съответните задачи се решават с помощта на компютър, в случая в среда на MATLAB.

С помощта на MATLAB едно обикновено диференциално уравнение (ОДУ) може да се реши както в затворена форма (когато това е възможно), така и числено. Численото решаване се налага поне в три случая: 1) явното решение съществува, намерено е от MATLAB, но е неудобно за пряко използване; такъв може да е случаят, когато решението се изразява чрез неелементарни функции; 2) решението в затворена форма е получено като неявна функция; 3) решение в затворена форма не съществува, или поне MATLAB не го намира.

Тук трябва да се има предвид и това, че при скаларни уравнения $\mathrm{d}y/\mathrm{d}t = f(t, y)$ решението чрез неявна функция, получено от MATLAB, може да е във вид, позволяващ изразяване на t чрез y (например, ако f не зависи от t). И накрая, MATLAB може да решава и уравнения, зададени неявно, например $y'^2 + y^2 = 1$.

Често е полезно уравнението да се реши и по двата начина: в затворена форма и числено. Решение в затворена форма се намира в MATLAB с командата `dsolve`. С тази команда се пресмятат и примитивни на функция. По-общо, ако f е зададена функция на t, то командата `F = dsolve('Dmy = f')`, където m е натурално число, пресмята в явен вид функция F, такава че $F^{(m)}(t) = f(t)$. Тук е поучително да се намерят примитивни F на прости функции f, които не се изразяват чрез елементарни функции, например когато $f(t) = \sin(t)/t$.

Числените методи често са основната техника за изследване на решенията на трудни математически проблеми. Във вградените функции на MATLAB са реализирани различни алгоритми, например методите на Ойлер, Рунге–Кута, Тейлър и др. Потребителят трябва да избере най-подходящата функция за задачата си в съответствие с нейната сложност и с желаната бързина и точност.

Преди да разгледаме решаването на ОДУ с помощта на MATLAB, представяме без доказателство минимума от теоретични сведения за съществуване и единственост на решението и за съответните числени методи за пресмятането му.

Нека е дадено векторното диференциално уравнение

$$y'(t) = f(t, y),\ t \in T = [a, b] \subset \mathbb{R},$$

с начално условие $y(t_0) = y_0$, $t_0 \in T$. Тук $y(t) \in \mathbb{R}^m$ е стойността на търсеното решение y в момента t, $f : T \times \mathbb{R}^m \to \mathbb{R}^m$ е зададена функция и $y_0 \in \mathbb{R}^m$ е зададено начално състояние. Предполагаме, че в околност на двойката (t_0, y_0) функцията $t \mapsto f(t, y)$ е непрекъсната при фиксирано y и удовлетворява условието на Липшиц

$$\|f(t, y_1) - f(t, y_2)\| \le L\|y_1 - y_2\|;\ |t - t_0| < \tau,\ \|y_k - y_0\| < \eta,$$

където $L = L(\tau, \eta) < \infty$ (накратко, но по-неточно: функцията $f(t, y)$ е непрекъсната по t и Липшицова по y). Тогава съществува единствено решение $y = \varphi(t; t_0, y_0)$, такова че $\varphi(t_0; t_0, y_0) = y_0$, определено в някакъв максимален интервал $J \subset T$, съдържащ точката t_0.

Полезно е да се покаже как нарушаването на условията на Липшиц води до нееднинственост на решението. За целта разглеждаме скаларното уравнение $y'(t) = 2\sqrt{y}$ при $y \ge 0$ и с начално условие $y(0) = 0$. Задачата има тривиално решение $y = 0$. Същевременно тя допуска и континуум от решения y_c, $c \ge 0$, определени от $y_c(t) = 0$ при $0 \le t \le c$ и $y_c(t) = (t - c)^2$ при $t > c$.

Командата за точно интегриране `dsolve` на ОДУ е със следния синтаксис:
`dsolve('eq1','eq2',...,','cond1','cond2',...)`. Тук с `eq1,eq2,...` са зададени уравненията, използвайки `t` като независима променлива, а с `cond1,cond2,...` са зададени начални и/или гранични условия. По подразбиране независимата променлива е t. Ако искаме да означим тази променлива с друга буква, например x, я пишем като последен аргумент, например
`dsolve('eq1','eq2',...,','cond1','cond2',...,'x')`. Така командата

```
>> y=dsolve('Dy=(1-2*x^2*y)/(x*(1+x^2))','x')
```

дава общото решение на линейното уравнение $y' = (1 - 2x^2 y)/(x(1 + x^2))$.

Чрез командата `dsolve` могат да се решават системи от диференциални уравнения без допълнителни условия (общо решение), както и системи с начални и/или гранични условия. Могат да се решават и уравнения с непълен набор от допълнителни условия, например N уравнения с $M < N$ начални и/или гранични условия. Тук е полезно да се покаже как прости гранични задачи нямат решение или допускат безбройно много решения. Например уравнението $y'' + y = 0$ с гранични условия $y(0) = 0$, $y(b) = B$ има единствено решение $y(t) = (B/\sin(b))\sin(t)$ когато числото b/π не е цяло, 2) има континуум от решения $y(t) = C\sin(t)$, $C \in \mathbb{R}$, когато b/π е цяло и $B = 0$ и 3) няма решения, когато b/π е цяло и $B \neq 0$. В първия случай MATLAB дава решение `b/sin(b)*sin(t)`, във втория – решение `C1*sin(t)` и в третия случай генерира съобщение `undefined`.

Числените методи позволяват приближено пресмятане на решението в точки от интервала $[a, b]$ на изменение на t. Тези методи работят добре, когато задачата е добре обусловена, т. е. когато малки изменения в данните – дясната страна и началните и/или гранични условия, водят до малки изменения в решението.

При численото решаване на ОДУ разглеждаме едностъпковите методи от типа на Рунге–Кута. За целта записваме уравнението като (векторно) уравнение от първи ред. Този процес наричаме *канонизиране*. Някои от функциите за числено интегриране на ОДУ са предназначени за решаване на диференциално-алгебрични системи от първи ред, при които част от уравненията са крайни (алгебрични), например $y' = f(t, y, z)$, $g(y, z) = 0$, където $y(t) \in \mathbb{R}^m$, $z(t) \in \mathbb{R}^n$ и $f : T \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$.

Необходимо е студентите да притежават минимум познания за числените методи. При едностъпковите методи избираме $N + 1$ точки t_k, където $a = t_0 < t_1 < \cdots < t_N = b$. В най-простия случай $t_{k+1} - t_k = h = (b-a)/N$. Стойността на приближеното решение в точката t_k означаваме с y_k, като очакваме да е изпълнено приближеното равенство $y_k \simeq y(t_k)$.

При най-простата явна схема на Ойлер имаме

$$y_{k+1} = y_k + hf_k,\ \ f_k = f(t_k, y_k);\ \ k = 0, 1, \dots, N - 1.$$

При неявната схема на Ойлер пресмятаме $z = y_{k+1}$ от уравнението $z = y_k + f(t_k, z)$ с няколко прости итерации $z^{[s+1]} = y_k + f(t_k, z^{[s]})$, $z^{[0]} = y_k$.

Две други, по-точни схеми за интегриране, са

$$y_{k+1} = y_k + hf(t_k + h/2, y_k + hf_k/2)$$

и

$$y_{k+1} = y_k + (h/2)(f_k + f(t_{k+1}, y_k + hf_k)).$$

Препоръчва се студентите да напишат (с наша помощ) програми на MATLAB за реализиране на тези четири схеми и да тестват тяхното изпълнение върху моделен пример с известно решение, например $y'(t) = Ay(t)$, $y(0) = 1$, при което $y(t) = \exp(At)$. Полезно е да се анализира грешката $|y_{N+1} - \exp(Ab)|$ при нарастване на N, включително до настъпване на "взрив" на грешката при твърде голямо N. Може да се разгледа и класическата схема на Рунге–Кута от 4 ред.

Описание на функциите за числено интегриране може да се намери първо с командата `help funfun` и после с `help xxx`, където `xxx` е наименованието на съответната функция. Поради отсъствие на адекватен термин на български е прието една такава функция да се нарича *солвър* (от английското *solver*). Всички солвъри се извикват по един и същ начин (с изключение на някои специални опции). В зависимост от вида на задачата може да се използват различни солвъри и да се сравнява тяхното действие.

Численото решаване на ОДУ става по следната схема.

1. Съставяме съответната система от диференциални или диференциално–алгебрични уравнения от първи ред.
2. Съставяме М–файл с описание на входните данни.
3. Викаме избрания солвър от командния прозорец на MATLAB.

С MATLAB могат да се решават и гранични задачи за системи от вида $y'(t) = f(t, y(t))$, за които търсим частно решение в интервал $[a, b]$, което да удовлетворява граничните условия $h(y(a), y(b)) = 0$. Както показахме, такава задача може да има единствено решение, да има множество решения или да няма решения. Поради това е полезно при числено решаване на тези задачи да се зададе подходящо начално приближение. Какви начални стойности ще зададем зависи от опита ни да решаваме гранични задачи или от характера на уравнението. Решенията на някои задачи могат да бъдат много чувствителни към началните приближения. Ако задачата има единствено решение, то можем да очакваме при всички начални приближения да получаваме един и същ резултат. Когато решаваме непозната задача, можем да опитаме с различни приближения. Ако се получи съобщение за грешка, значи сме задали недопустими стойности за задачата. Когато граничната задача има много решения, избираме това, което е от практически интерес.

Библиотечка с М–файлове за решаване на примерни ОДУ може да се намери на адрес `http:/www.uacg.bg/books/matlab-ode`.

Заключение. В тази работа описахме използването на системата MATLAB за решаване на някои задачи от аналитичната геометрия, както и за решаване на ОДУ. У нас университетите нямат единна практика при използването на програмни математически системи. В някои университети се използва системата Maple, която по отношение на разглежданите задачи има близки възможности до тези на MATLAB [4]. Време е да унифицираме практиката по тези въпроси и да предприемем единни действия с МОМН за купуване на лицензирани копия за всички студентски работни места в държавните университети.

ЛИТЕРАТУРА

[1] Й. Йорданов. MATLAB: Преобразувания, изчисления, визуализация. Техника, София, 2005 [ISBN 954-03-0659-0].

[2] M. Konstantinov. Foundation of Numerical Analysis (with MATLAB Examples). UACEG Publ., Sofia, 2007, PDF text available at `http://www.uacg.bg/books/math/ma-new.pdf`.

[3] C. Moler. Numerical Computing with MATLAB. The MathWorks, Inc., 2004, PDF text available at `http://www.mathworks.com/moler`.

[4] В. Пашева, Я. Арнаудов. Основи на числените методи. Изд. ТУ–София, София, 2002 [ISBN 954-90427-5-8].

Михаил Константинов
Владимир Тодоров
Галина Пелова
Юлиана Бонева
Факултет по математика
Университет за архитектура, строителство и геодезия
1046 София, България
e-mail: mmk_fte@uacg.bg
vtt_fte@uacg.bg
galina_fte@uacg.bg
boneva_fte@uacg.bg

USING MATLAB IN TECHNICAL UNIVERSITIES

Mihail Konstantinov, Vladimir Todorov, Galina Pelova, Juliana Boneva

We consider some aspect of the mathematical education in technical universities using MATLAB. Special attention is paid to the subjects Analytical Geometry and Differential Equations.