ИЗПОЛЗВАНЕ НА КОМПЮТЪРНАТА ПРОГРАМА “СМЕТАЛО” ЗА ЧИСЛЕНО ИНТЕГРИРАНЕ НА ДИФЕРЕНЦИАЛНИ УРАВНЕНИЯ

доц. д-р Деко Деков
Великотърновски университет “Св. св. Кирил и Методий”

The use of the computer program “Smetalo” for numerical integration of differential equations

Dr. Deko Dekov
University of Veliko Turnovo

Abstract

The computer program “Smetalo”, build by the author, is a useful tool for mathematical computations. The program, version 1.02 was published on the CD of the journal “PC Magazine Bulgaria” in November 2002, and the version 1.036 was published on the CD of the same journal in May 2003. Here we show how the program can be used for numerical integration of differential equations.

Keywords: computer program, differential equations, method of Euler.

Компютърната програма “Сметало”, създадена от автора, е полезен инструмент за решаване на широк кръг задачи. Програмата работи с операционната система “Windows”. Програмата е създадена на английски език и локализирана на български език. Версия 1.02 на програмата на български език е публикувана в CD на списание “PC Magazine Bulgaria”, ноември 2002 г. Версия 1.036 е публикувана в CD на същото списание през май 2003 г. По този начин програмата е достъпна на широк кръг потребители в България. Версия на програмата на български език може да бъде изтеглена от Интернет, например от адреса http://smetalo.tripod.com.

Програмата “Сметало” може да бъде ползувана от потребители, незапознати с програмирането. В този случай се въвеждат входните данни и при щракване върху бутона “Отговор” програмата дава отговора. По този начин могат да бъдат получени отговорите на широк кръг задачи, групирани в направления “Алгебра”, “Геометрия”, “Линейна алгебра”, “Комбинаторика”, “Информатика”, “Финанси”.

Програмата може ефективно да бъде използувана и от потребители, запознати с програмирането. Програмата включва работен лист (“worksheet”), който включва текстова област за код, бутони “Отговор” и “Изчисти” и текстова област за изход. При писане на код в текстовата област за код и щракане върху бутона “Отговор”, в текстовата област за изход веднага се вижда резултата.

По този начнн програмата е удобен инструмент както за бързо и лесно усвояване на програмирането, така и ефективен инструмент за бързо и лесно решаване на широк кърг задачи. В програмата са дадени многобройни примери за решаване на задачи посредством писане на програми. Тези програми се пишат бързо и лесно, имат малък обем, а дават отлични резултати.

Тук ще разгледаме още един пример за използване на програмата “Сметало”, а именно съставянето и използването на програма за числено интегриране на обикновени диференциални уравнения с метода на Ойлер.

Задачата за числено интегриране на обикновени диференциални уравниня е една от основните изчислителни задачи в математиката. Тя се изучава от студентите от всички технически специалности. Едно отлично изложение е дадено в известния учебник на Пискунов [2].

Нека напишем програма за числено интегриране на обикновени диференциални уравнения с метода на Ойлер. Текста на програмата е следният:

```
function f(x,y)
{
  return x + y;
}
a = 0; b = 1; n = 10; h = (b - a) / n;
x = 0; y = 1; z = "";

for (i = 1; i <= n; i++)
{
  y = f(x,y) * h + y;
  x += h;
  tx = x;
  tx = tx.toFixed(1);
  ty = y;
  ty = ty.toFixed(10);
  z += i + "  y(" + tx + ") = " + ty + "\n";
}
z;
```

Нека поясним кода на програмата. Интегрираме обикновено диференциално уравнение от вида y’=f(x,y). Първо дефинираме функцията f(x,y). В горния пример сме положили f(x,y) = x+y. След това декларираме променливите, които са необходими и ги инициализираме, задавайки като стойности на тези променливи данните за конкретната задача, която решаваме. Тези променливи са следните: a и b - границите на интервала, в който интегрираме, n – брой на стъпките, т.е. броя на частите, на които разделяме интервала [a,b], x и y – за задаване на задачата на Коши, т.е. началната точка с координати (x,y), през която трябва да минава търсената интегрална крива. С цикъл for извършваме пресмятанията, със свойството toFixed(m) указваме колко цифри след десетичната запетая (в компютърните програми -десетична точка), да бъдат използвани, а с променливата z указваме какво да съдържа и как точно да изглежда изхода. Стойността на променливата, която е записана последна, е изход на програмата.

За тестване на програмата беше избран учебника на Пискунов [2]. В този учебник е разгледан следният пример: интегриране на диференциалното уравнение y’=x+y в интервала [0,1] при начално условие y(0)=1 и при 10 стъпки. Данните в програмата по-горе съответстват на примера, разгледан в Пискунов [2]. При щракване върху бутона “Отговор” програмата дава следния изход:

```
1   y(0.1) = 1.1000000000
2   y(0.2) = 1.2200000000
3   y(0.3) = 1.3620000000
4   y(0.4) = 1.5282000000
5   y(0.5) = 1.7210200000
6   y(0.6) = 1.9431220000
7   y(0.7) = 2.1974342000
```

```
8   y(0.8) = 2.4871776200
9   y(0.9) = 2.8158953820
10  y(1.0) = 3.1874849202
```

Забелязва се от третата стъпка нататък несъответствие между резултатите на програмата и резултатите, дадени в учебника на Пискунов [2]. За да видим къде е несъответствието, правим малка промяна в програмата, така че изхода да бъде таблица като тази, дадена в учебника на Пискунов [2]. Правим и малка добавка, която дава абсолютната и относителната грешка. Имената на функции и променливи могат да бъдат писани на български език. Получаваме програмата:

```
function f(x,y)
{
  return x + y;
}
a = 0; b = 1; n = 10; h = (b - a) / n;
x = 0; y = 1; z = "";

for (i = 1; i <= n; i++)
{
  y = f(x,y) * h + y;
  x += h;
  tx = x;
  tx = tx.toFixed(1);
  ty = y;
  ty = ty.toFixed(4);
  u = x + y;
  u = u.toFixed(4);
  v = (x + y) * h;
  v = v.toFixed(4);
  z += i + " " + tx + " " + ty + " " + u + "    " + v +"\n";
}

    точна_стойност = 2 * Math.exp(b) - b - 1;
    точна_стойност = точна_стойност.toFixed(10);
    абсолютна_грешка = Math.abs(y - точна_стойност);
    абсолютна_грешка = абсолютна_грешка.toFixed(6);
    относителна_грешка = абсолютна_грешка / точна_стойност;
    относителна_грешка *= 100;
    относителна_грешка = относителна_грешка.toFixed(3);
    z += "\nточна стойност = " + точна_стойност +"\n"
       + "абсолютна грешка = " + абсолютна_грешка + "\n"
       + "относителна грешка = " + относителна_грешка + " % ";
```

която дава следния изход:

```
1    0.1   1.1000   1.2000   0.1200
2    0.2   1.2200   1.4200   0.1420
3    0.3   1.3620   1.6620   0.1662
4    0.4   1.5282   1.9282   0.1928
5    0.5   1.7210   2.2210   0.2221
```

```
6    0.6   1.9431   2.5431   0.2543
7    0.7   2.1974   2.8974   0.2897
8    0.8   2.4872   3.2872   0.3287
9    0.9   2.8159   3.7159   0.3716
10   1.0   3.1875   4.1875   0.4187

точна стойност = 3.4365636569
абсолютна грешка = 0.249079
относителна грешка = 7.248 %
```

Може лесно да се види, че грешката в таблицата на Пискунов [3] е при събирането в третия ред. Това влече погрешност на резултатите до края на таблицата, а също така погрешност на абсолютната и относителната грешки, дадени в учебника. Програмата, дадена по-горе се пише бързо и лесно и с едно щракване мишката получаваме резултатите. Ако пресмятанията бъдат извършени без компютър, ще е необходимо време и освен това може да бъде допусната грешка, както показва таблицата в учебника на Пискунов [2].

Същата програма може да бъде използвана за числено интегриране на произволно уравнение от вида y'=f(x,y). Трябва само да запишем данните на конкретната задача. По този начин лесно могат да бъдат решени практически всички задачи, дадени в учебници и сборници, отнасящи се до интегриране на обикновени диференциални улавнения с метода на Ойлер. Задачите, включени в учебни пособия обаче по необходимост са с малък брой пресмятания, тъй като се предполага, че студентите ще ги пресмятат без кампютър. Ние обаче можем лесно да запишем с клавиатурата като брой на стъпките не 10, както е в пограмата по-горе, а например 1000. При записване на 1000 абсолютната и относителната грешки намаляват с по около 100 пъти, както ни осведомява програмата.

Като пример, да решим задача 4103 от сборника на Берман [1]. Условието на задачата се вижда от данните, които сме въвели, като единствената разлика е, че сме указали на компютъра да смята с 10 точни цифри след десетичната запетая, а не с 2, както е в сборника на Берман [1]. След въвеждане на данните на задачата получаваме:

```
function f(x,y)
{
  return x*y*y*y + x*x;
}
a = 0; b = 1; n = 20; h = (b - a) / n;
x = 0; y = 0; z = "";

for (i = 1; i <= n; i++)
{
  y = f(x,y) * h + y;
  x += h;
  tx = x;
  tx = tx.toFixed(2);
  ty = y;
  ty = ty.toFixed(10);
  z += i + "  y(" + tx + ") = " + ty + "\n";
}
z;
```

При щракване с мишката върху бутона “Отговор” получаваме следния изход:

```
1   y(0.05) = 0.0000000000
2   y(0.10) = 0.0001250000
3   y(0.15) = 0.0006250000
4   y(0.20) = 0.0017500000
5   y(0.25) = 0.0037500001
6   y(0.30) = 0.0068750007
7   y(0.35) = 0.0113750056
8   y(0.40) = 0.0175000313
9   y(0.45) = 0.0255001385
10  y(0.50) = 0.0356255116
11  y(0.55) = 0.0481266420
12  y(0.60) = 0.0632547074
13  y(0.65) = 0.0812623002
14  y(0.70) = 0.1024047403
15  y(0.75) = 0.1269423265
16  y(0.80) = 0.1551440363
17  y(0.85) = 0.1872934069
18  y(0.90) = 0.2236976333
19  y(0.95) = 0.2647013620
20  y(1.00) = 0.3107073340
```

Подобно на горната програма за числено интегриране на диференциялни уравнения с метода на Ойлер, лесно и бързо могат да бъдат написани програми за числено интегриране на диференциални уравнения с други метода, освен метода ва Ойлер, а също така програми за числено интегриране на системи диференциални уравнения.

Лесни за писане и малки по обем са програмите за числено пресмятане на определени интеграли – с метода на правоъгълниците, на трапеците, на Симпсон. Тази тема е широко застъпена в учебните програми.

За студенти, незапознати с програмирането, може да бъде изготвен интерфейс с текстови полета, в които да бъдат въвеждани входните данни. Програмата може да бъде снабдена с възможността да начертае изучаваната интегрална крива (или няколко интегрални криви). Освен на студентите, програмата може да служи и на преподавателите – за лесно и бързо съставяне на нови задачи.

Литература

1. Берман, Г. Н., Сборник задач по курсу математического анализа, Наука, Москва, 1985 (20-е издание).
2. Пискунов, Н. С., Диференциальное и интегральное исчисления, том 2, Наука, Москва 1978 (12-е издание).