

Н. А. Стахин

## ПРИМЕР ИСПОЛЬЗОВАНИЯ КОМПЬЮТЕРНОЙ АЛГЕБРЫ МАХИМА В ДИСЦИПЛИНЕ «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ»

Показано, что использование системы компьютерной математики Махима качественно улучшает рассмотрение темы «Свободное падение с учетом сопротивления среды», традиционно изучаемой в дисциплине «Компьютерное моделирование». Махима позволяет получить аналитическое решение нелинейного дифференциального уравнения, аналитические выражения для скорости падения и пройденного пути, найти асимптотическое поведение скорости падения и визуализировать решение.

**Ключевые слова:** модель, компьютерное моделирование, компьютерная алгебра Махима, встроенный язык программирования, оператор цикла, дифференциальное уравнение, начальные условия, задача Коши, свободное падение, сила тяготения, сила сопротивления, сила Архимеда, закон Стокса, коэффициент лобового сопротивления.

Широко известно, что моделирование является эффективным, мощным орудием познания. По этой причине учебная дисциплина «Компьютерное моделирование» входит в число базовых учебных дисциплин, преподаваемых в современных вузах, готовящих специалистов-исследователей, а основы компьютерного моделирования начинают изучать уже в старших классах общеобразовательной средней школы. Методику преподавания дисциплины «Компьютерное моделирование» изучают студенты педагогических университетов. Традиционно наряду с физической постановкой задач моделирования в университетском курсе компьютерного моделирования изучаются численные методы решения задач, такие как численное интегрирование, численное дифференцирование, численное решение трансцендентных уравнений и другие, в то время как аналитические компьютерные методы дисциплиной не востребованы. Хотя современные компьютеры знакомы не только с арифметикой, но и с алгеброй.

Компьютеры XXI в. умеют производить аналитические преобразования, дифференцировать и интегрировать и решать дифференциальные уравнения не только численно, но и аналитически, в виде формул. И, кроме того, что немаловажно при моделировании, пакеты компьютерной математики содержат встроенные графические средства, позволяющие визуализировать решения, получать таковые сразу в виде графиков.

В настоящей работе, не отменяя традиционное изложение, авторы покажут методику органичного внедрения компьютерной алгебры Махима в конкретном примере рассмотрения темы «Свободное падение тела с учетом сопротивления среды», традиционно рассматриваемой в курсе компьютерного моделирования.

В школьном курсе физики сопротивлением среды пренебрегают, даже не объясняя причин.

В классическом труде Могилева, Пака и Хеннера [1], переиздаваемом на протяжении 15 лет, тема «Свободное падение тела с учетом сопротивления среды» рассматривается подробно и всесторонне. Но классическое рассмотрение темы только выигрывает, если наряду с используемым в книге численным методом решения задачи практическое применение получит компьютерная математика и ее аналитические способности.

В работе Могилева и соавт. [1] рассмотрение темы «Свободное падение тела с учетом сопротивления среды» построено очень грамотно: на первом этапе записываются уравнения движения свободно падающего тела. Рассматривается движение шарика в воздухе, воде и глицерине. На втором этапе моделируется эпизод из кинофильма «Небесный тихоход», в котором герой фильма майор Булочкин падает с высоты 6 тыс. м в реку без парашюта.

Следуя труду Могилева и соавт. [1], силу сопротивления со стороны среды будем моделировать суммой двух слагаемых, зависящих от скорости падения  $v$ :

$$F_{\text{сопр}} = k_1 v + k_2 v^2, \quad (1)$$

где  $k_1, k_2$ , – постоянные коэффициенты. Коэффициент  $k_1$ , определяется свойствами среды и формой тела. Для шарика  $k_1 = 6\pi\mu r$  согласно закону Стокса, где  $\mu$  – динамическая вязкость среды,  $r$  – радиус шарика. Коэффициент  $k_1$  пропорционален площади поперечного сечения тела  $S$ , плотности среды  $\rho_{\text{среды}}$  и зависит от формы тела через безразмерный коэффициент лобового сопротивления  $c$  в виде  $k_2 = \frac{1}{2} c S \rho_{\text{среды}}$ .

Силу притяжения к Земле запишем в виде произведения массы тела  $m$  и ускорения свободного падения  $g$ , точно так же, как и в работе [1],

$$F_{\text{тяж}} = mg, \quad (2)$$

где  $g$  считаем не зависящим от высоты, постоянным.

Записав второй закон Ньютона – ускорение пропорционально силе и обратно пропорционально массе, для скорости падения тела  $v$  находим дифференциальное уравнение вида

$$\frac{dv}{dt} = \frac{mg - k_1 v - k_2 v^2}{m}, \quad (3)$$

тождественное уравнению (7.7) работы [1].

В отношении (7.7) авторы [1] сетуют на то, что даже в очень простой задаче возникла необходимость в решении дифференциального уравнения нестандартного типа. Они допускают, что с помощью хитроумных подстановок, имеющихся в справочнике Камке, возможно, кому-то удастся найти решение дифференциального уравнения в виде суперпозиции алгебраических и трансцендентных функций. Однако авторы [1] полагают, что после нахождения скорости падения как функции времени  $v = v(t)$  шансы найти в квадратурах пройденный путь  $s$  как интеграл от скорости

$$s(t) = \int_0^t v(t) dt \quad (4)$$

будут «уже совсем невелики» (см. текст и формулу (7.9) в работе [1]).

Мы со своей стороны полагаем, что нам справочник Камке не понадобится и наши шансы резко возрастут, если для решения поставленных задач использовать систему компьютерной математики (СКМ) *Math* [2–5].

Среди других СКМ пакет *Math* характеризуется тем, что является бесплатной, многоплатформенной, активно развивается, является свободно распространяемой системой с открытым кодом, не требует для работы дополнительных ресурсов компьютера, а по качеству работы ничем не уступает наиболее известным проприетарным системам *Maple* и *Mathematica*. Дистрибутив *Math* русифицирован и свободно скачивается с сайта разработчиков программы [6]. Однако *Math* пока не имеет справки на русском языке, а техническая документация на английском достаточно трудна для начинающих, поэтому рассматриваемый в настоящей работе пример снабжен подробными поясняющими комментариями.

Будем решать далее в *Math* дифференциальное уравнение вида

$$v' + av + bv^2 = g, \quad (5)$$

где  $v$  – функция (скорость падения), зависящая от аргумента (времени)  $t$ ,

$v'$  – производная по времени,

$a, b, g$  – положительные константы

$$a = \frac{k_1}{m}, \quad b = \frac{k_2}{m}, \quad (6)$$

где  $k_1, k_2$  – коэффициенты выражения (1), используемые при записи зависимости силы сопротивления от скорости.

В компьютерной алгебре *Math* решение обыкновенных дифференциальных уравнений находит функция

`ode2 (Уравнение, функция, аргумент),`

в качестве первого аргумента которой следует записать дифференциальное Уравнение, далее указать имя функции и имя аргумента. Для решения уравнения (5) команду (задание) для *Math* запишем в виде (%i1) (см. лист. 1). Используемую команду предваряет комментарий (текст, находящийся между скобками /\* \*/), содержание которого никак не влияет на исполнение команды.

#### Листинг 1

```
(%i1) /* находим общее решение */
ode2('diff(v,t,1)+a*v+b*v^2=g,v,t);

Is 4 b g + a^2 positive or negative? p;

log( (-2 b v + sqrt(4 b g + a^2) - a) /
      (2 b v + sqrt(4 b g + a^2) + a) )
(%o1) - ----- = t + %c
      sqrt(4 b g + a^2)
```

Наличие в уравнении (5) производной первого порядка мы указали в виде 'diff(v, t, 1). При поиске решения (%o1) *Math* спросила, является ли выражение положительным или оно отрицательное (Is positive or negative?).

В качестве ответа на заданный вопрос мы указали  $p$  (positive) и получили общее решение дифференциального уравнения – довольно страшное на вид выражение (%o1), содержащее скорость и в числителе, и в знаменателе, под знаком логарифма, и содержащее произвольную аддитивную константу %c в правой части решения. Из вида решения становится понятен вопрос, заданный *Math* по поводу выражения  $4bg + a^2$ . Из этой величины *Math* извлекла квадратный корень и результат поместила в знаменатель.

Для нахождения частного решения, соответствующего начальному условию  $v = 0$  при  $t = 0$  (задача Коши, отвечающая случаю нулевой начальной скорости падения), необходимо использовать функцию

`ic1 (Общее решение, начЗн аргумента, начЗн функции).`

При этом нет необходимости преобразовывать формулу (%o1) в строковый формат для детальной записи первого аргумента (общее решение) для функции ic1 (), но можно сослаться на него как на % – последний полученный результат, согласно принятому в Maxima способу (лист. 2).

Листинг 2

```
(%i2) /* подставляем начальные условия
находим частное решение */
ic1(% , t=0, v=0);
log(
  -2 b v + sqrt(4 b g + a^2) - a
  -----
  2 b v + sqrt(4 b g + a^2) + a
)
(%o2) -----
sqrt(4 b g + a^2)
sqrt(4 b g + a^2) t - log(
  2 b g
  -----
  a sqrt(4 b g + a^2) + 2 b g + a^2
)
-----
sqrt(4 b g + a^2)
```

Maxima сумела найти решение задачи Коши, отвечающей нулевой начальной скорости. Она записала частное решение в виде формулы (%o2), которая уже не содержит произвольной константы %c. Однако решение (%o2) представляет собой уравнение, в котором функция v и аргумент t связаны неявно. Поскольку нас более устраивает явная зависимость функции от своего аргумента (так как далее мы желаем интегрировать скорость по времени, чтобы получить пройденный путь), мы тут же попросили Maxima решить алгебраическое уравнение (%o2) относительно переменной v.

Команда (%i3) может быть записана вручную или может быть вызвана через пункт меню программы Уравнение, Решить, ответственного за вызов функции solve (). Синтаксис использования функции solve () имеет вид:

```
solve ([список уравнений], [список переменных]).
```

В команде (%i3) для Maxima мы опять не стали выписывать список уравнений, который в нашем случае содержит только одно уравнение, а сослались на [список уравнений] как на ранее полученный результат [%o2].

Листинг 3

```
(%i3) /* решаем относительно v */
solve([%], [v]);
(%o3) [v =
  (g sqrt(4 b g + a^2) + a g) %e sqrt(4 b g + a^2) t - g sqrt(4 b g + a^2) - a g
  -----
  (a sqrt(4 b g + a^2) + 2 b g + a^2) %e sqrt(4 b g + a^2) t + 2 b g
```

Maxima решила (%o2) относительно функции v и получила в явном виде зависимость скорости падения от времени  $v = v(t)$ . Формально решение (%o3) представляет собой список найденных ре-

шений. Признаком списка в Maxima служат квадратные скобки. Первый (и единственный) элемент списка (%o3) – это уравнение вида  $v = v(t)$ , но интересующий нас результат – это правая часть уравнения. Чтобы формально получить только правую часть решения  $v(t)$ , достаточно обратиться к переменной v, но добавить ссылку % [1] – на первый элемент только что полученного списка (который в данном случае можно конкретизировать также как %o3 [1]). Следующая команда (см. лист. 4): (%i4) v,% [1] позволяет избавиться от формальных признаков списка – квадратных скобок [], знака равенства = и левой части уравнения, содержащей идентификатор функции v.

Листинг 4

```
(%i4) /* записываем выражение для скорости */
v,%[1];
(%o4) (g sqrt(4 b g + a^2) + a g) %e sqrt(4 b g + a^2) t - g sqrt(4 b g + a^2) - a g
-----
(a sqrt(4 b g + a^2) + 2 b g + a^2) %e sqrt(4 b g + a^2) t + 2 b g
```

Выражение (%o4) может быть использовано для расчета скорости в любой произвольный момент времени, для нахождения пройденного пути и построения графика.

Путь s можно рассчитать согласно (4) как определенный интеграл от скорости, где аргумент t изменяется от 0 – нижнего предела интегрирования до некоторого  $t = T$  – верхнего предела интегрирования в полном согласии с формулой (7.9) работы [1].

Вычисление интегралов в СКМ Maxima выполняет функция

integrate (Подынтегральное выражение, аргумент, нижПредел, верхПредел).

После интегрирования путь, пройденный за T секунд, оказывается представлен аналитическим выражением %o5

Листинг 5

```
(%i5) /* интегрируем скорость и получаем путь */
integrate(% , t, 0, T);
Is T positive, negative or zero? p;
(%o5) - (-2 log( abs( a sqrt(4 b g + a^2) %e sqrt(4 b g + a^2) T +
(2 b g + a^2) %e sqrt(4 b g + a^2) T + 2 b g )) + sqrt(4 b g + a^2) T +
a T + 2 log( a sqrt(4 b g + a^2) + 4 b g + a^2 )) / (2 b)
```

И здесь при вычислении определенного интеграла необходимо пояснить для Maxima, что используемый новый параметр T – число положительное (p – positive).

Необходимо отметить, что, рассматривая полученное выражение для пройденного пути %o5, трудно увидеть, что при больших значениях времени  $T$  путь изменяется линейно и движение происходит с постоянной скоростью. Однако вывод о равномерном движении нетрудно получить из выражения %o4 (см. лист. 4), рассматривая асимптотическое поведение скорости падения при  $t \rightarrow \infty$ . Легко видеть, что при  $t = \infty$  выражение %o4 представляет собой неопределенность вида  $\frac{\infty}{\infty}$ , поскольку при  $t = \infty$  экспонента вида  $\%e^{\sqrt{4bg+a^2}t}$  принимает бесконечное значение.

Используя знания по теории пределов стандартного курса математического анализа, нетрудно раскрыть неопределенность и записать предельное значение скорости в виде отношения коэффициента при экспоненте, находящейся в числителе, к коэффициенту при такой же экспоненте, находящейся в знаменателе. Но можно не записывать ответ, найденный умозрительно, хотя и вполне очевидный, а поручить раскрытие стандартного предела СКМ *Maxima*, как это сделано на лист. 6.

**Листинг 6**

```
(%i6) /* предел скорости при t = inf */
limit(((g*sqrt(4*b*g+a^2)+a*g)*
%e^(x)-g*sqrt(4*b*g+a^2)-a*g)/
((a*sqrt(4*b*g+a^2)+2*b*g+a^2)*
%e^(x)+2*b*g),x,inf);
(%o6) 
$$\frac{g\sqrt{4bg+a^2}+ag}{a\sqrt{4bg+a^2}+2bg+a^2}$$

```

Здесь неопределенность вида  $\frac{\infty}{\infty}$  раскрыла и получила предельную скорость функция СКМ *Maxima*

`limit (Выражение, переменная, значПеременной)`, возвращающая предел *Выражения*, для значения переменной, стремящейся к значению *значПеременной*. Бесконечное значение  $\infty$  переменной на языке *Maxima* указано как *inf*.

Чтобы облегчить *Maxima* нахождение предела, в команде %o6 мы ввели временную переменную  $x = \%e^{\sqrt{4bg+a^2}t}$ .

*Выражение* для скорости в команде %i6 не писалось вручную, тем более что оно громоздкое и при записи нетрудно ошибиться, а был использован интерфейс wx*Maxima*, позволяющий конвертировать строковый формат выражения в математическую запись и обратно. При щелчке мыши на ма-

тематической записи %o4 последняя оказывается выделенной. Если теперь скопировать выделение (Ctrl+C) в буфер обмена, то сразу же после вставки (Ctrl+V) скопированного его можно редактировать как обычный текст. Заменить в данном случае сложное выражение вида на простую переменную, например  $x$ . Легко видеть, что предел скорости в виде аналитического выражения %o6 *Maxima* рассчитала как частное двух коэффициентов при экспонентах.

Покажем, что предельная скорость  $\check{v}$ , которая была рассчитана в работе [1] как решение алгебраического уравнения, тождественна выражению %o6. Следуя труду [1], приравняем нулю производную скорости по времени  $v' = 0$  в уравнении (5), получим квадратное уравнение

$$av + bv^2 = g. \tag{7}$$

Равенство нулю производной скорости означает равномерное движение с постоянной скоростью. Иными словами: физический смысл утверждения (7) соответствует равенству силы сопротивления силе тяжести.

Найдем решение уравнения (7) с помощью функции `solve ()` и покажем, что положительное решение уравнения при делении его на предельную скорость (выражение %o6) тождественно единице – лист. 7.

**Листинг 7**

```
(%i7) /* ищем решение уравнения */
solve(a*x+b*x^2=g,x);
(%o7) [x = - $\frac{\sqrt{4bg+a^2}+a}{2b}$ ,
x =  $\frac{\sqrt{4bg+a^2}-a}{2b}$ ]
(%i8) /* берём положительный корень*/
x,%[2];
(%o8)  $\frac{\sqrt{4bg+a^2}-a}{2b}$ 
(%i9) /* упрощаем частное*/
radcan(%/%o6);
(%o9) 1
```

Таким образом, решения всех задач, сформулированных в теме «Свободное падение...», компьютерная алгебра *Maxima* получает в аналитическом виде.

Рассчитаем численные значения коэффициентов дифференциального уравнения (5) для случая, рассматриваемого в книге: моделирования падения летчика – майора Булочкина – с высоты 6 тыс. м. Падение происходит в воздухе, используем те же,

что и в работе [1], численные значения для физических констант:

$$g = 9,81 \frac{\text{М}}{\text{с}^2}; c = 1,22; S = 0,7 \text{ м}^2,$$

$$\rho_{\text{среды}} = 1,29 \frac{\text{КГ}}{\text{М}^3}; m = 80 \text{ КГ}.$$

И возьмем нулевое значение для коэффициента  $a = \frac{k_1}{m} = 0$ , поскольку линейной составляющей силы сопротивления при расчетах в работе [1] пренебрегли – см. лист. 8. Кроме того, рассчитаем численно предельную скорость падения. С этой целью командой `%об` вызовем выражение для предельной скорости с добавлением опции `numer`.

**Листинг 8**

```
(%i10) /* Для майора Булочкина,
a=0, m=80 кг, g=9.81,
c=1.22, S=0.7, ro=1.29 */

a: 0; g: 9.81;
b: 1/2*1.22*0.7*1.29/80;
%об,numer;

(%o10) 0
(%o11) 9.810000000000001
(%o12) 0.006885375
(%o13) 37.74597991911805
```

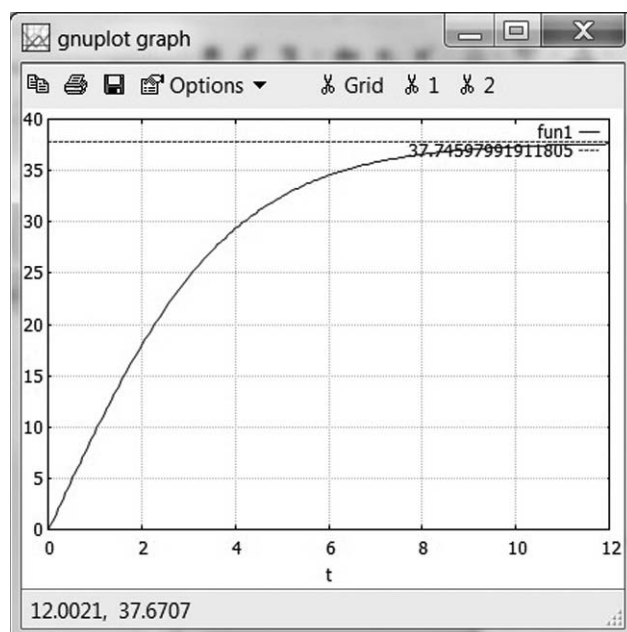
Теперь `Maxima` может автоматически подставлять значения  $g = 9,81$ ;  $a = 0$ ;  $b = 0,006885375$  в формулы для скорости и пройденного пути `%o4` и `%o5` для каждого требуемого значения времени.

Предельная скорость падения, рассчитанная по формуле `%об`, составила  $37,75 \text{ м/с}$ , что вполне согласуется с соответствующим значением  $37,76 \text{ м/с}$ , полученным в работе [1] численным методом типа Рунге – Кутта второго порядка, называемым также исправленным методом Эйлера.

Зависимость скорости падения от времени представлена на рисунке, созданном в СКМ `Maxima`.

На рис. 1 представлены сразу два графика: один из них выходит из начала координат и показывает изменение скорости падения со временем. Второй график (нарисован пунктиром) представляет собой горизонтальную прямую, соответствующую предельной скорости. Для построения графиков использован интерфейс `gnuplot graph`, который имеет многочисленные собственные варианты настройки и имеет небольшой сервис для визуального анализа графика: при смещении мыши вдоль графика в левом нижнем углу можно видеть координаты положения мыши.

Команда для визуализации скорости падения имеет очень простой вид:



Изменение скорости со временем

`wxplot2d ([%o4,%o13], [t,0,15], [gnuplot_preamble, "set grid;"]) $,`

где ссылка `%o4` отсылает на формулу, содержащую зависимость скорости от времени, а ссылка `%o13` отсылает на численное значение предельной скорости.

Значения скорости и пройденного пути для промежуточных значений времени удобно получить в виде таблицы (лист. 9), где значения величин даются с округлением до четырех значащих цифр.

**Листинг 9**

```
t= 1  v= 9.595  s= 4.851
t= 2  v= 18.03  s= 18.8
t= 3  v= 24.63  s= 40.29
t= 4  v= 29.36  s= 67.43
t= 5  v= 32.52  s= 98.48
t= 6  v= 34.55  s= 132.1
t= 7  v= 35.81  s= 167.3
t= 8  v= 36.58  s= 203.6
t= 9  v= 37.05  s= 240.4
t= 10 v= 37.33  s= 277.6
t= 11 v= 37.5  s= 315.0
t= 12 v= 37.6  s= 352.6
t= 13 v= 37.66 s= 390.2
t= 14 v= 37.69 s= 427.9
t= 15 v= 37.71 s= 465.6
(%o16) done
```

Рассчитанные в СКМ значения (лист. 9) сравнивались с соответствующими величинами из табл. 7.2 работы [1], рассчитанными численным методом. Среднеквадратичное расхождение по скорости составило 2% (результаты, полученные численным методом, систематически занижены).

Текст задания для *Maxima* (%i16), с помощью которого были получены результаты лист. 9, весьма громоздкий и нуждается в обширных комментариях, поэтому приведено задание отдельно, в текстовом виде (лист. 10), и ниже расположено описание алгоритма.

#### Листинг 10

```
(%i16)
block ([i], W: makelist (((g*sqrt(4*b*g + a^2) + a*g) *%e^
(sqrt(4*b*g + a^2) *t) - g*sqrt(4*b*g + a^2) - a*g) / ((a*
sqrt(4*b*g+a^2) + 2*b*g + a^2) *%e^ (sqrt(4*b*g + a^2)
*t) + 2*b*g), t, 1, 15), Z: makelist (- (-2*log(abs(a*sqrt
(4*b*g + a^2) *%e^ (sqrt(4*b*g + a^2) *T) + (2*b*g + a^2)
*%e^ (sqrt(4*b*g + a^2) *T) + 2*b*g)) + sqrt(4*b*g+a^2)
*T + a*T + 2*log(abs(a*sqrt(4*b*g + a^2) + 4*b*g +
a^2))) / (2*b), T, 1, 15), for i:1 thru 15 do (fpprintprec: 4, d:
float(W [i]), f: float(Z [i]), print (« t=», i,» v=», d,» s= «, f))
```

Команда (%i16) содержит операторы, объединенные в блок, и использует элементы встроенного языка программирования: цикл типа `for` вызывает функцию `makelist ()` и оператор вывода на печать `print`. В блоке описана локальная переменная `i`, которая служит параметром цикла. Вместо обычной ссылки на результат вида (%o4) здесь полностью оказывается необходим текст формулы для скорости, поскольку зависимость от параметра в функции `makelist ()` передается явно через выражение. Функция `makelist ()` необходима для создания списка значений, подлежащих выводу на печать.

Функция `makelist ()` генерирует два списка: список значений скоростей `W` и список значений пройденного пути `Z`. Список `W` содержит 15 элемен-

тов – скоростей падения для 15 значений времени в интервале 1–15 с шагом 1. Аналогичный список `Z` содержит такое же количество значений пройденного пути `s` для тех же самых значений времени из интервала 1–15. Синтаксис использования функции `makelist ()` следующий

`makelist (Выражение, Параметр, первоеЗнач, последнееЗнач).`

Для вывода на печать элементов списка используется цикл вида

`for i: 1 thru 15 do (Блок вывода на печать).`

Чтобы ограничить число значащих цифр, выводимых на печать, глобальной переменной компьютерной алгебры `fpprintprec` внутри блока присвоенное значение 4, а при выводе используются временные переменные `d` и `f`, значения которым передаются с помощью функции `float`.

В заключение попутно сделаем замечание по поводу легитимности исследуемого дифференциального уравнения (3). Уравнение (3) было записано как уравнение движения свободно падающего тела (уравнение (7.7) работы [1]). Однако при выводе этого уравнения никак не учитывалась сила Архимеда – выталкивающая сила, создаваемая средой, направленная вверх. Архимедова сила пропорциональна объему тела и плотности среды. При падении в воздухе ею можно пренебречь ввиду низкой плотности воздуха. Однако в случае воды и глицерина архимедовой силой пренебрегать нельзя, так как в случае воды и глицерина деревянный шар будет не тонуть, а всплывать, потому что сила Архимеда в этих случаях превышает силу тяжести. При падении стального и медного шариков через воду и глицерин пренебрегать силой Архимеда также не следует ввиду сравнительно высоких плотностей жидких сред. По недосмотру в книге [1] сила Архимеда из внимания упущена, поэтому оценка предельных скоростей в случае воды и глицерина там выполнена неверно и нуждается в корректировке, начиная с записи уравнения движения.

#### Список литературы

1. Могилев А. В., Пак Н. И., Хеннер Е. К. Информатика: учеб. пособие для студ. пед. вузов. М.: Академия, 1999–2012. 816 с.
2. Чичкарев Е. А. Компьютерная математика с *Maxima*: руководство для школьников и студентов. М.: Alt Linux, 2012. 384 с.
3. Чичкарев Е. А. Академия Alt Linux: компьютерная математика с *Maxima*. URL: <http://www.intuit.ru/studies/courses/3484/726/info/> (дата обращения: 01.09.2015).
4. Маевский Е. В., Ягодковский П. В. Компьютерная математика. Высшая математика в СКМ *Maxima*. Ч. I. Введение. М.: Финансовый университет, 2014. 196 с. URL: <http://e-math.ru/maxima> (дата обращения: 01.09.2015).
5. Стахин Н. А. Основы работы с системой аналитических (символьных) вычислений *Maxima*: учебное пособие. М., 2008. 86 с. URL: <ftp://ftp.altlinux.ru/pub/people/black/MethodBooks/Maxima.pdf> (дата обращения: 01.09.2015).
6. Сайт разработчиков *Maxima*. URL: <http://sourceforge.net/projects/maxima/files/> (дата обращения: 01.09.2015).

Стахин Н. А., кандидат физико-математических наук, доцент.  
**Томский государственный педагогический университет.**  
Ул. Киевская, 60, Томск, Россия, 634061.  
E-mail: Stakhin@tspu.edu.ru

Материал поступил в редакцию 21.05.2015.

*N. A. Stakhin*

### AN EXAMPLE OF USING COMPUTER ALGEBRA MAXIMA IN THE DISCIPLINE “COMPUTER MODELLING”

The article deals with the modelling of pilot falling from a height of 6 000 meters in water. It is shown that the use of computer algebra system (CAS) Maxima qualitatively improves the consideration of the topic “Freefall considering environmental resistance”, traditionally studied in the discipline “Computer simulation”. Maxima has found the general solution of the nonlinear differential equation and found a partial solution with zero initial conditions – the Cauchy problem. CAS Maxima has found the analytical expression for the rate of fall and height fallen down as the functions of time. CAS Maxima has obtained the asymptotic behavior of the rate of fall in the time and has calculated the rate limit of fall. CAS Maxima calculated numerically intermediate values of fall speed and height fallen down for the 15 time values in the range of 1–15 in increments of 1. These values are compared with analogical values calculated by numerical method in book, with the standard deviation for rate is equal 2%. CAS Maxima has plot graphics the rate of fall versus the time.

**Key words:** *model, computer modeling, computer algebra Maxima, integrated programming language, cycle operator, differential equation, initial conditions, Cauchy problem, free-fall, force of gravity, force of resistance, Archimedes force, Stokes' law, drag coefficient.*

### References

1. Mogilev A. V., Pak N. I., Henner E. K. *Informatika. Uchebnoye posobiye dlya studentov pedagogicheskikh vuzov* [Informatics. Tutorial book for students of pedagogical universities]. Moscow, Akademiya Publ., 1999–2012. 816 p. (in Russian).
2. Chichkarev E. A. *Komp'yuternaya matematika s Maxima. Rukovodstvo dlya shkol'nikov i studentov* [Computer mathematics with Maxima: A guide for schoolchildren and students]. Moscow, Alt Linux Publ., 2012. 384 p. (in Russian).
3. Chichkarev E. A. *Akademiya Alt Linux: Komp'yuternaya matematika s Maxima* [Academy of Alt Linux: Computer mathematics with Maxima]. URL: <http://www.intuit.ru/studies/courses/3484/726/info/> (accessed 1 September 2015) (in Russian).
4. Mayevskiy E. V., Yagodovskiy P. V. *Komp'yuternaya matematika. Vysshaya matematika v SKM Maxima. Chast' 1. Vvedeniye. Uchebnoye posobiye* [Computer mathematics. Higher mathematics CAS Maxima. Part I. Introduction: Manual]. Moscow, Finansovyy universitet Publ., 2014. 196 p. URL: <http://e-math.ru/maximal/> (accessed 1 September 2015) (in Russian).
5. Stakhin N. A. *Osnovy raboty s sistemoy analiticheskikh (simvol'nykh) vichisleniy Maxima: uchebnoye posobiye* [Basics of the work with system of analytical (symbolic) computations in Maxima: Textbook]. Moscow, 2008. 86 p. URL: <ftp://ftp.altlinux.ru/pub/people/black/MethodBooks/Maxima.pdf> (accessed 1 September 2015) (in Russian).
6. *Sayt razrabotchikov Maxima* [Website of Maxima developers] URL: <http://sourceforge.net/projects/maxima/files/> (accessed 1 September 2015).

Stakhin N. A.  
**Tomsk State Pedagogical University.**  
Ul. Kievskaya, 60, Tomsk, Russia, 634061.  
E-mail: Stakhin@tspu.edu.ru