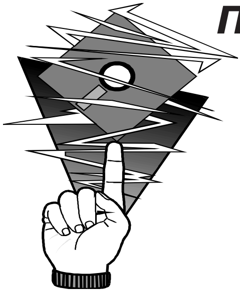

ПРОГРАММНО-ТЕХНИЧЕСКИЕ СРЕДСТВА

А.А. Черняк, доктор ф.-м.н., профессор кафедры математики Белорусского государственного педагогического университета им. М.Танка,
А. Якимович, студент физического факультета Белорусского государственного педагогического университета им. М.Танка

Методика применения Maple в образовательном процессе вуза

Современный образовательный процесс в вузе характеризуется высокой интенсивностью – большим объемом сообщаемой информации в единицу времени, что предъявляет особые требования к методике построения учебного процесса по отдельным дисциплинам. В частности, повышение эффективности обучения в современных условиях невозможно без использования систем компьютерной математики (СКМ), которые освобождают учебный процесс от трудоемких и неэффективных расчетов, построения таблиц и графиков вручную, и позволяют преподавателю сконцентрировать основные усилия на постановке задачи, выборе метода ее решения, интерпретации результатов решения [1].

В последние годы появились учебные пособия по различным математическим (и не только) дисциплинам с использованием пакета MathCAD [2-7], который на сегодняшний день является одним из обязательных компонентов компьютерных технологий на всех уровнях образовательной системы Республики Беларусь и России. Популярность этого пакета объясняется его универсальностью, относительной легкостью изучения и дружественным интерфейсом. Благодаря встроенным возможностям для численных и символьных вычислений и решения различных типов уравнений и неравенств, эффективность MathCAD особенно высока при изучении общего курса высшей математики, математического программирования, численных методов, корреляционного и регрессионного анализа, экономико-математических моделей. Однако операторы программирования в MathCAD, хотя и содержат основные конструкции языков высокого уровня, скорее ориентированы на усвоение начинающими пользователями СКМ общих принципов алгоритмизации, чем на разветвленное искусное программирование, без которого невозможно решение сложных прикладных задач, возникающих при изучении целого ряда дисциплин и, в частности, теории вероятностей.

Справедливости ради отметим, что ранее уже предпринимались попытки изучения теории вероятностей с использованием MathCAD (например, в пособии [6], опирающемся на классический учебник Фелера). Однако, фактически, все они сводились к искусственному внедрению в учебный процесс громоздких и плохо структурированных программных модулей, которые не способны составить конкуренцию традиционным, но более доступным и наглядным способам иллюстрации и интерпретации классических понятий теории вероятностей.

В качестве альтернативы нами разработан комплекс программных модулей в интегрированной среде Maple, позволяющих:

а) отойти от рассмотрения упрощенных задач, сводящихся к простому оперированию формулами, и перейти к более сложным и приближенным к практике экспериментальным расчетам (см. пример 1);

б) посредством компьютерного моделирования дать «экспериментальную» мотивацию и пропедевтику ключевых определений и понятий теории вероятностей, тем самым сделав их доступными и естественными для изучения (см. пример 2);

в) предвосхитить практически все важнейшие классические результаты теории вероятностей, обеспечив визуализацию изучаемых закономерностей (см. пример 3).

При этом от студентов не требуется знания инструментария Maple. В то же время они без труда смогут модифицировать программные модули, опираясь на элементарные навыки программирования, приобретенные при изучении языков высокого уровня. Более того, открываются широкие возможности ненавязчивого обучения системе Maple (хотя бы и на начальном уровне) посредством дополнительных заданий по написанию простейших приложений, расширяющих рамки применения уже имеющихся модулей (см. пример 1).

Продемонстрируем на конкретных примерах упомянутые выше методические принципы проведения лабораторно-практических занятий на базе СКМ Maple.

Пример 1. При изучении условного математического ожидания используется понятие мартингала. На упрощенном «игровом» языке этот термин тождественен понятию «справедливой игры» («fair game»), в которой условное ожидаемое значение $M(S_n \mid S_{n-1} = a_{n-1}, K, S_1 = a_1)$ случайной величины S_n, выражающей выигрыш в n-м раунде, равно ее значению a_{n-1} в (n-1)-м раунде. Имеются игры, установить «справедливость» которых аналитическими методами весьма проблематично. В подобных случаях на помощь приходит компьютерное моделирование. В качестве колоритного примера рассмотрим упрощенную модель игры на бирже акций.

Предположим, что стоимость акции ежедневно увеличивается на один доллар с вероятностью 0,5 или уменьшается на один доллар с вероятностью 0,5. Предположим также, что на бирже маклер Алиса придерживается следующей стратегии. В первый день торгов она покупает акцию по цене V. Затем она ждет, когда цена акции увеличится на 1$ до величины $V + 1$, и затем продает эту акцию. В последующие дни она продолжает наблюдать за ходом торгов до того дня, когда цена акции опять упадет до величины V. В этот день она снова покупает акцию и ждет, пока ее цена не подрастет до величины $V + 1$, чтобы затем ее продать. Очевидно, Алиса может проиграть только в том случае, если в последний день торгов она все еще будет держателем акции, цена которой так и не достигнет величины $V + 1$.

Программный модуль *StockPlay* (рис. 1) моделирует возможный ход событий в течение n дней (без потери общности считается, что $V = 0$). Программа вычисляет прибыль Алисы по истечению торгов, а также выводит точки с координатами $(i,f(i))$, где i пробегает целые значения от 0 до n, а $f(i)$ равно стоимости акции в i-й день торгов. Прерывистыми линиями показаны периоды, в течение которых Алиса была держателем акции, сплошными линиями – периоды, в течение которых у нее акции не было.

```
> StockPlay:=proc(n)
   local i,j,U,V,plotlist,own,pointlist,temp,a,linelist,temppoint,
   prevown,benefits:
   with(plottools): with(plots):
   U:=rand(0..1):
   V:=0:
   benefits:=0:
   own:=true:
   plotlist:=[]:  linelist:=[]:
   pointlist:=[[0,0]]:
   for i from 1 to n do
    prevown:=own:
    temp:=V:
    if U()=1 then V:=V+1: else V:=V-1:
    fi:
    for j from 1 to 10 do
     pointlist:=[op(pointlist),[i-1+j/10,temp+(V-temp)*j/10]]:
    od:
    if V=1 then own:=false fi:
    if V=0 then own:=true fi:
    if (not own and prevown) then benefits:=benefits+1 fi:
    if not prevown then
     linelist:=[op(linelist),line([i-1,temp],[i,V])]
    fi:
    if (own and not prevown) or (i=n and not prevown) then
     plotlist:=[op(plotlist),op(linelist)]:
     linelist:=[]:
    fi:
   od:
```

Рис. 1

```
a:=plot(pointlist,style=point, symbol = POINT, color = 'black'):
plotlist:=[op(plotlist),a]:
if own then benefits:=benefits+V fi:
lprint(`         benefits =`,benefits);
display(plotlist);
end:
> StockPlay(30);
```

```
benefits =   4
```

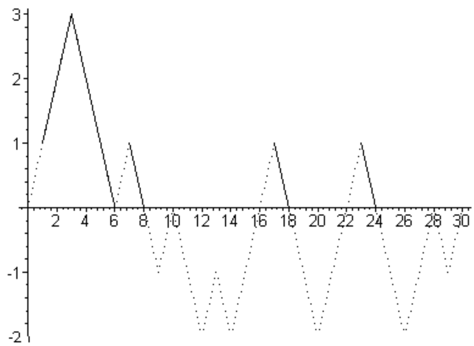

Рис. 1 (продолжение)

Студентам можно предложить доработать данный модуль с тем, чтобы он выполнялся (без вывода графика) достаточно большое число раз с одновременным подсчетом ожидаемой средней прибыли и последующим построением полигона частот, с которыми Алиса может получить ту или иную прибыль. При этом будет установлено, что несмотря на близкую к нулю ожидаемую среднюю прибыль (и в этом случае игра является «справедливой»), вероятность получения положительной прибыли, тем не менее, значительно превышает 0,5. Таким образом, можно будет сделать вывод: выбранная система «игры на бирже» дает больше шансов для получения прибыли, чем простое хранение купленной акции, а «справедливая игра» часто остается таковой даже в условиях субъективного выбора стратегии.

Пример 2. В самом начале изучения непрерывных вероятностных распределений вводятся ключевые понятия плотности распределения $f(x)$ и вероятностной функции $P(X \in E)$, где $E \subset \Omega$, Ω – непрерывное пространство исходов, X – непрерывная случайная величина. Предлагаем провести «экспериментальную» пропедевтику и мотивацию этих понятий следующим образом.

Рассмотрим опыт: бросание дротика на круг радиусом 1. Если нас интересует точность броска – расстояние от центра круга, то пространством исходов будет множество $\Omega = [0;1]$. Отметим на мишени некоторую область E и промоделируем бросание дротика на компьютере с помощью программного модуля *DartProb*. Убедимся, что частота попаданий в область E приблизительно равна отношению площади E к площади всей мишени, равной π. Свяжем с данным опытом непрерывную случайную величину X, выражающую расстояние упавшего дротика от центра мишени. Поступим в соответствии с результатами компьютерного моделирования и положим $P(X \le a) = \frac{\pi a^2}{\pi} = a^2$. Отсюда

$$P(a \le X \le b) = P(X \le b) - P(X \le a) = b^2 - a^2.$$

Эту же формулу можно записать и по-другому с помощью функции

$$f(x) = \begin{cases} 2x, & \text{если } x \in [a;b] \\ 0, & \text{если } x \notin [a;b] \end{cases}$$

следующим образом: если $E = [a;b]$, то

$$P(X \in E) = P(E) = \int_E f(x)dx = \int_a^b 2xdx = b^2 - a^2.$$

Чем же обусловлена догадка взять в качестве функции $f(x)$ именно функцию $2x$?

Разобьем единичный круг на m концентрических колец одинакового радиуса $1/m$ и промоделируем n бросков дротика на этот круг с помощью модуля *DartDens* (рис. 2). Результатом выполнения модуля будет гистограмма, в которой площадь i-го прямоугольника равна частоте попадания дротика в i-е концентрическое кольцо.

```
DartDens:=proc(n,show)
 local result,count,U,x,y,distance, m:
 with(stats):
 result:=[]: count:=1:
 U:=random[uniform[-1,1]]:
 while count<(n+1) do
  x:=U():  y:=U():
  distance:=sqrt(x^2 + y^2):
  if distance<=1 then
   result:=[op(result),distance]:
   count:=count+1:
  fi:
 od:
 if show then print(result) fi;
  m:=10: Gistogram(result,0,1,m);
end:
DartDens(8000,false);
```

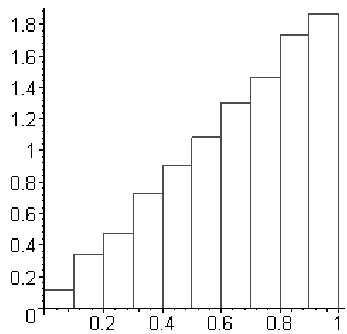

Рис. 2.

На рисунке 2 показан модуль *DartDens* и результат его выполнения при $m = 10$, $n = 8000$ (здесь опущена вспомогательная процедура построения гистограмм). Из рисунка видно, что «ступеньки» гистограммы располагаются вдоль прямой $y = 2x$.

Таким образом, приходим к следующим определениям. Вероятностной функцией $P(E)$, заданной на множестве событий E пространства исходов Ω, называется функция $P(E)$, для которой можно указать такую неотрицательную функцию $f(x)$, отвечающую условию нормировки $\int_{-\infty}^{+\infty} f(x)dx = 1$, что $P(E) = \int_E f(x)dx$ для любого события $E \subset \Omega$. При этом $f(x)$ называется плотностью распределения непрерывной случайной величины X, если $P(X \in E) = P(E) = \int_E f(x)dx$ для любого события $E \subset \Omega$.

Пример 3. Следующий модуль *CLTGraph* (рис. 3), иллюстрирующий центральную предельную теорему, дает возможность студентам, экспериментируя с исходными данными, лучше понять природу основного закона теории вероятностей.

В списке distributionlist задается ряд распределения дискретной случайной величины X. Затем с помощью вспомогательных модулей вычисляются математическое ожидание $M(X)$ и дисперсия $D(X)$ случайной величины X; определяется случайная величина S_n, равная сумме n независимых случайных величин X (здесь эти процедуры опущены). Затем строятся кривая плотности стандартного нормального распределения и «костыльная» гистограмма нормализованной случайной величины $\frac{S_n - n \cdot M(X)}{\sqrt{n \cdot D(X)}}$ с корректирующим коэффициентом $\frac{1}{\sqrt{n \cdot D(X)}}$ вдоль оси Oy. На рисунке 3 видно, что уже распределение суммы десяти одинаковых независимых случайных величин X хорошо приближает нормальное распределение.

Литература

1. Черняк А.А., Доманова Ю.А., Ранько Т.Н. Синтез классической и компьютерной математики в обучении // Информатизация образования. – № 1. – 2005. – С. 36-45.

```
CLTGeneral1Plot:=proc(n,distributionlist)
 local mu,sigma2,data,data2,difflist,g,i,j,plot1,plot2,E:
 with(plots):
 E := evalf(exp(1)):
 data:=[]:  difflist:=[]:  data2:=[]:
 mu:=DiscreteMean(distributionlist):
 sigma2:=DiscreteVariance(distributionlist):
 data:=ConvolutionN(distributionlist,n):
 data:=sort(data):
 for i from 2 to nops(data) do
  difflist:=[op(difflist),op(1,op(i,data))-op(1,op(i-1,data))]:
 od:
 g:=igcd(op(difflist)):
 for j from 1 to nops(data) do
  data2:=[op(data2),[(op(1,op(j,data)) - (n * mu))/sqrt(n * sigma2),
           op(2,op(j,data)) * sqrt(n * sigma2)/g]]:
 od:
 plot1:=Spikegraph(data2,-4,4,true);
 plot2:=plot((1/sqrt(2*Pi))*(E^((-x^2)/2)),x=-4..4):
 display([plot1,plot2]);
end:
CLTGeneral1Plot(10,[[1,.4],[2,.3],[3,.1],[4,.1],[5,.1]]);
```

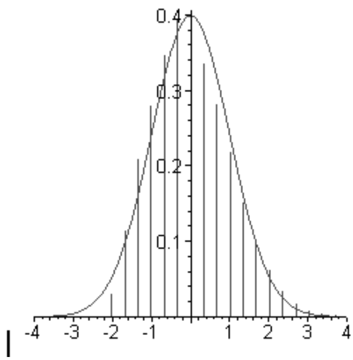

Рис. 3.

2. Плис А.И., Сливина. MathCAD 2000: математический практикум для экономистов и инженеров. – М.: Финансы и статистика, 2000. – 655 с.

3. Черняк А.А., Мельников О.И., Кузнецов А.В. Математика для экономистов на базе MathCAD. – СПб: БХВ, 2003. – 485 с.

4. Салманов О.Н. Математическая экономика с применением MathCAD и Excel. – СПб: БХВ, Петербург, 2003. – 464 с.

5. Черняк А.А., Доманова Ю.А., Черняк Ж.А. Высшая математика на базе MathCAD. Общий курс. – СПб: БХВ, 2004. – 593 с.

6. Андронов А.М., Копытов Е. А., Гринглаз Л. Я. Теория вероятностей и математическая статистика. – СПб: Питер, 2004. – 464 с.

7. Поршнев С.В., Беленкова И.В. Численные методы на базе MathCAD. – СПб: БХВ, 2005. – 450 с.