

ИННОВАЦИОННЫЕ ПЕДАГОГИЧЕСКИЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

О НЕКОТОРЫХ ОСОБЫХ СЛУЧАЯХ ИСПОЛЬЗОВАНИЯ ЧИСЛЕННЫХ МЕТОДОВ В РЕШЕНИИ МАТЕМАТИЧЕСКИХ ЗАДАЧ ПРИМЕНИТЕЛЬНО К РЕАЛИЗАЦИИ ОБУЧАЮЩИХ ПРОГРАММНЫХ ПРОДУКТОВ ПО МАТЕМАТИКЕ

А.П. Возженников

Кафедра высшей математики и программного обеспечения ЭВМ
Мурманский государственный технический университет
ул. Спортивная, 13, Мурманск, Россия, 183010

Рассмотрены особые случаи использования численных методов в разработке обучающих программных продуктов: модификация или компиляция известных методов, привлечение статистических методов в качестве альтернативы аналитическим формулам, дискретизация значений функции; приведены примеры использования, описаны алгоритмы и математические модели.

Вычислительный эксперимент как индуктивный подход, используемый часто для нахождения решения сложных математических задач, может также использоваться применительно к классу учебных задач при условии их реализации на ЭВМ. Возможностью построения математической модели для автоматизации решения представленной задачи во многом определяется степень ее вариативности. Основное отличие учебной задачи от научной заключается в том, что преподаватель всегда может упростить модель фиксированием значений некоторых ее параметров либо поставить существенные ограничения на параметры модели. Это позволяет утверждать, что для всякой учебной математической задачи возможно построить модель с такими ограничениями на значения параметров этой модели, что задача станет несложной для программной реализации, а адекватность ее модели может быть подтверждена вычислительным экспериментом на достаточно большом наборе учебных задач. В качестве таких ограничений могут выступать точность и диапазон допустимых значений параметров модели, степень надежности получаемых результатов и другие ограничения.

Программная реализация подобных моделей позволяет не только решать математические задачи учебного назначения, но и автоматизировать процесс исследования решений, динамически визуализировать объекты, а также прогнозиро-

вать их поведение. Это позволяет пользователю (обучающемуся) приобрести некоторые базовые навыки работы с объектами изучения: вычисление значений, построение графиков, определение характеристик, исследование влияния параметров и т.п. Реализация автоматического решения задач по конкретной теме предоставляет также возможность организации модулей самопроверки и тестирования путем генерации задач со случайными значениями параметров из предусмотренного множества их значений и последующей автоматической проверкой ответов [1].

Очевидно, что подобные разработки требуют привлечения математического аппарата, несколько более сложного, чем тот, который используется в рамках изучаемой дисциплины при решении задач вручную. Опыт разработки обучающих программных продуктов поддержки учебного процесса по дисциплинам «Математический анализ» и «Высшая математика» позволяет выделить в программной реализации отдельных приложений некоторые особые случаи использования численных методов, которые описаны ниже.

Модификация или компиляция известных методов. Теоретически известный численный метод решения задачи часто требует адаптации к предметной области. Такая адаптация может повлечь за собой как обобщение метода, так и его функциональное ограничение. Возможны случаи объединения нескольких методов, т.е. компиляция.

В качестве примера реализации подхода рассмотрим определение численных характеристик функции при ее исследовании, в том числе предельного поведения функции. Для численного нахождения нулей функции на заданном отрезке необходимо расширить классический метод, предполагающий разницу знаков функции на концах отрезка, до более общего, который будет включать проверку существования значения функции в каждой из проверяемых точек отрезка, а также выделять случаи, когда функцию можно считать равной нулю на некоторой части отрезка.

Таким образом, можно предположить следующий алгоритм нахождения нулей функции на заданном отрезке:

1) задается функция $f(x)$, концы отрезка a и b , точность вычислений ε и шаг дискретизации h для значений аргумента x ;

2) вычисляются значения функции $f(x)$ в каждом узле одномерной сетки, покрывающей отрезок $[a, b]$;

3) при смене знака функции (при выполнении условия $f(x-h) \cdot f(x) < 0$) используются текущее x и предыдущее $(x-h)$ значения аргумента как исходные данные для применения классического численного метода нахождения нуля на промежутке $(x-h, x)$ с заданной точностью ε . Отметим, что одним из универсальных в этом случае является метод половинного деления промежутка (метод бисекции) как обладающий свойством глобальной сходимости;

4) в случае близости значения функции $f(x)$ к нулю с заданной точностью (при выполнении условия $|f(x)| < \varepsilon$) запомнить позицию;

5) если для следующих n вычисленных значений функции $f(x+h)$, $f(x+2h)$, ... $f(x+n \cdot h)$, выполняется то же условие ($|f(x)| < \varepsilon$), то считать функ-

цию $f(x)$ близкой к нулю на данном промежутке $[x; x + nh]$; в случае невыполнения этого условия считать первую точку x нулем функции.

Если в ходе вычислений учитывать, в каких точках функция не может быть вычислена, поскольку не существует, можно также использовать описанный метод для нахождения области определения функции на отрезке. Такую возможность предоставляет классический метод структурной обработки исключительных ситуаций, имеющийся в любой среде программирования.

Описанный расширенный вариант реализации автоматического исследования свойств функции легко переносится на исследование свойств первой и второй производной этой функции, которые могут быть вычислены конечно-разностными методами. Такая компиляция двух численных методов позволяет автоматизировать процесс исследования функции вплоть до получения области ее существования, численных значений нулей функции, критических точек и возможных точек перегиба. Такой подход исключает определение промежуточных результатов, а только проверяет численные ответы.

Привлечение статистических методов в качестве альтернативы аналитическим формулам. Анализ некоторых характеристик объекта предполагает исследование этого объекта в целом, при любых значениях его параметров. Автоматизация подобного анализа невозможна, если исходить из неопределенности понятий «бесконечность» и «любой» относительно программной реализации проверки некоторых определений или свойств. Однако установить различие или отсутствие определенного свойства объекта возможно вероятностным обобщением характеристик объекта, полученных для некоторой случайной выборки значений его параметров. Разумеется, такой подход не является математически строгим, но вполне оправдывает себя при автоматизации решений некоторых учебных задач.

Примером реализации этого подхода может быть определение аналитических характеристик функции в исследовании ее свойств. Так, четность (или нечетность) функции предполагает выполнение условия $f(x) = f(-x)$ (или $f(x) = -f(x)$) для всех точек множества задания функции, симметричного относительно нуля. При достаточно большом объеме выборки случайных значений аргумента вероятность того, что условие четности (или нечетности) выполнится также для любых других точек, близка к единице. Аналогично можно установить свойство однородности функции нескольких переменных $g(t \cdot x, t \cdot y) = t^n g(x, y)$, проверив выполнение этого условия для двумерной выборки, сформированной случайным образом из точек заданной области.

При таком решении возникает проблема точного сравнения результатов вычисления двух функций при одинаковых значениях аргументов. Поэтому при сравнении вычисленных значений функций используется приближенное совпадение с заданной точностью ε .

В рамках программной реализации учебных задач, связанных с вычислением кратных интегралов, эффективным оказалось использование метода Монте-Карло. Применение этого метода для вычисления определенного интеграла приведено в работе [2].

Рассматривается функция $g(x)$, непрерывная на промежутке $[a, b]$. Требуется приближенно вычислить значение определенного интеграла:

$$I = \int_a^b g(x) dx \quad (1)$$

Вводится непрерывная случайная величина ξ , определенная на промежутке $[a, b]$ плотностью распределения $p(x)$, причем функция $p(x)$ может быть выбрана произвольно с обычными для плотности распределения условиями: $p(x) \geq 0$

при $\forall x \in [a; b]$ и $\int_a^b p(x) dx = 1$. Если ввести еще одну непрерывную случайную

величину $\eta = g(\xi) / p(\xi)$, то ее математическое ожидание совпадает со значением искомого интеграла (1):

$$M_\eta = \int_a^b \frac{g(x)}{p(x)} \cdot p(x) dx = I. \quad (2)$$

Далее рассматриваются N одинаковых случайных величин $\eta_1, \eta_2, \dots, \eta_N$, которые являются реализациями случайной величины η в N независимых испытаниях и имеют такие же математическое ожидание и дисперсию, как и случайная величина η . Если число N достаточно велико, то, согласно центральной предельной

теореме, распределение суммы $\sum_{j=1}^N \eta_j$ является близким к нормальному с пара-

метрами $a = N \cdot M_\eta = N \cdot I$, $\sigma^2 = N \cdot D_\eta$. В соответствии с правилом «трех сигм» для нормального распределения имеем

$$P \left\{ \left| \frac{1}{N} \sum_{j=1}^N \eta_j - I \right| < 3 \sqrt{\frac{D_\eta}{N}} \right\} \approx 0,997. \quad (3)$$

Это соотношение означает, что при достаточно большом значении N верно приближенное равенство:

$$I \approx \frac{1}{N} \sum_{j=1}^N \eta_j, \quad (4)$$

где $\eta_j = \frac{g(\xi_j)}{p(\xi_j)}$.

Соотношение (3) показывает также, что с вероятностью $P \approx 0,997$ ошибка приближения (4) не превосходит $3 \sqrt{D_\eta / N}$, т.е. обратно пропорциональна \sqrt{N} . Для расчета можно использовать любую случайную величину ξ , определенную на промежутке $[a, b]$, однако дисперсия D_η , а с ней и оценка погрешности метода зависят от того, какая случайная величина ξ будет использована.

Исходя из того, что задача является учебной и реализуется программно, целесообразно ограничиться для всех случаев одной и той же равномерно рас-

пределенной псевдослучайной величиной ξ , которую можно получить в любой современной среде программирования без привлечения дополнительных алгоритмов генерации псевдослучайных чисел. Такой подход существенно упрощает программную реализацию вычислений.

Для равномерного распределения $p(x) = \frac{1}{b-a}$ формула (4) имеет вид:

$$\int_a^b g(x)dx \approx \frac{b-a}{N} \sum_{j=1}^N g(\xi_j). \quad (5)$$

Видоизменим формулу (5), разделив обе части приближенного равенства на $(b-a) \neq 0$:

$$\frac{1}{b-a} \int_a^b g(x)dx \approx \frac{1}{N} \sum_{j=1}^N g(\xi_j) \quad (6)$$

Здесь левая часть содержит формулу для вычисления среднего значения подынтегральной функции $g(x)$, непрерывной на отрезке $[a, b]$, а правая часть эмпирически определяет среднее значение той же функции. Следовательно, можно утверждать, что значение определенного интеграла от функции $g(x)$ по отрезку $[a, b]$ приближенно равно среднему эмпирическому значению функции $g(x)$ на отрезке $[a, b]$, умноженному на длину этого отрезка l_{ab} .

$$\int_a^b g(x)dx \approx l_{ab} \cdot \frac{1}{N} \sum_{j=1}^N g(x_j). \quad (7)$$

Расширим это утверждение для кратных интегралов: значение кратного интеграла от функции $g(M)$ по области D приближенно равно среднему эмпирическому значению функции $g(M)$ в области D , умноженному на меру этой области (D — конечная область в пространстве R^n ; M — точка в области D). Для двойных и тройных интегралов получим формулы:

$$\iint_D g(x, y) dx dy \approx S_D \cdot \frac{1}{N} \sum_{j=1}^N g(x_j, y_j), \quad (8)$$

$$\iiint_{\Omega} g(x, y, z) dx dy dz \approx V_D \cdot \frac{1}{N} \sum_{j=1}^N g(x_j, y_j, z_j). \quad (9)$$

Если область D задается в декартовых координатах системой неравенств, то мера этой области вычисляется так:

$$S_D = \int_a^b (y_2(x) - y_1(x)) dx, \quad (10)$$

$$V_D = \int_a^b dx \int_{y_1(x)}^{y_2(x)} (z_2(x, y) - z_1(x, y)) dy. \quad (11)$$

Для вычисления интегралов (10) и (11) также может использоваться метод Монте-Карло.

Алгоритм генерации псевдослучайных точек $M_j(x_j, y_j, z_j)$, равномерно распределенных в области D , легко реализуется стандартной функцией Random при условии, что известны наибольшие и наименьшие значения независимых переменных: $a \leq x \leq b$, $c \leq y \leq d$, $e \leq z \leq f$:

- 1) получить случайную точку x_j на отрезке $[a, b]$;
- 2) получить случайную точку y_j на отрезке $[c, d]$;
- 3) получить случайную точку z_j на отрезке $[e, f]$;
- 4) если $M_j \in D$, вычислить в ней значение функции $g(M_j)$;

5) подсчитать количество точек N' , попавших в область D и положить в формулах (8) и (9) $N = N'$.

Приближенные наибольшее и наименьшее значения y и z могут быть получены перебором значений функций, ограничивающих область D , с заданным шагом h .

Серия вычислительных экспериментов на примерах из сборника задач по курсу математического анализа [3] показала, что относительная погрешность δ_I не превышает 1% при $N > 10^3$, вне зависимости от вида функции $g(M)$ и области интегрирования D . Эмпирически установлено, что вычисление меры области D точным методом (вручную) не оказывает существенного влияния на погрешность результатов, но усложняет алгоритмизацию метода. Результаты вычислительных экспериментов приведены в таблицах 1 и 2.

Таблица 1

$\int_0^\pi dx \int_0^x \cos(x+y)dy = -2$			
Кол-во точек N	Приближенное значение I	Абсолютная погрешность ΔI	Относительная погрешность δ_I
4 975	-2,0098	0,0098252	0,49126%
4 955	-2,0101	0,0101044	0,50522%
5 011	-2,0015	0,0014862	0,07431%

Таблица 2

$\int_0^1 dx \int_0^{1-x} dy \int_0^{1-x-y} \frac{1}{(x+y+z+1)^3} dz = \frac{1}{2} \left(\ln 2 - \frac{5}{8} \right)$			
Кол-во точек N'	Приближенное значение I	Абсолютная погрешность ΔI	Относительная погрешность δ_I
5 021	0,034185	0,000104592	0,306958%
4 992	0,034212	0,000065919	0,193463%
4 983	0,034291	0,000048584	0,142587%

Полученные формулы эффективны для применения в учебных задачах, связанных с численным интегрированием функций одной и нескольких переменных.

Дискретизация значений функции одной и нескольких переменных. Визуализация решения математических задач на компьютере изначально предполагает некоторую дискретность, обусловленную заданным масштабом изображения. Иными словами изначально существует определенный уровень детализации объектов, превышение которого не является целесообразным — разрешение экрана, то есть количество точек, отображаемых на мониторе (количество пикселей). Таким образом, всякое непрерывное множество может быть представлено как дискретное с единичным шагом, соответствующим изменению экранной координаты на один пиксель монитора. Исходя из этого, многие задачи возможно решать переборным способом для конечного числа дискретных значений функции и ее аргумента.

Общий подход заключается в следующем: имеется некоторая двумерная область, отображаемая на экране дискретным двумерным множеством пикселей. Каждый пиксель можно считать узлом сетки, покрывающей данную область. Шаг сетки соответствует изменению экранной координаты на один пиксель монитора. Требуется проверить в каждом узле сетки некую характеристику функции, заданной на этой области и графически отобразить эту характеристику на экране. Последовательный перебор всех узлов сетки с вычислением требуемой характеристики в каждом узле и закрашиванием соответствующего узлу пикселя даст искомое графическое решение.

Так, программная реализация задачи построения двумерной области по системе неравенств является элементарной при дискретизации значений переменных в заданных неравенствах. Перебирая все отображаемые на экране точки дискретной области с единичным шагом по каждой координате, можно проверить в каждой точке (x_j, y_j) выполнение всех неравенств системы для ее координат x и y . Если выделить каждую точку, удовлетворяющую системе неравенств, цветом, отличным от цвета области, то в результате перебора получается графическое представление искомой области.

Другим примером использования дискретизации может служить построение линии уровня функции $f(x, y)$ в заданной двумерной области D , также основанное на последовательном вычислении значения функции в каждом из узлов сетки, покрывающей область D . Если в некоторой точке (x_j, y_j) значение функции $f(x_j, y_j)$ отличается от заданного значения уровня $f(x, y) = C$ не более, чем на ε , то точку (x_j, y_j) считать принадлежащей линии уровня и закрашивать цветом. Следует отметить, что величина ε будет прямо пропорционально влиять на толщину линии уровня. Кроме того, толщина линии уровня будет одинаковой только на тех участках, где значения частных производных функции $f(x, y)$ будут мало отличаться по значению, что обеспечивает примерно одинаковую скорость изменения функции. Это значит, что в общем случае построенная таким способом линия уровня будет неравномерной толщины, что нежелательно в научных или производственных целях, но вполне приемлемо в учебных.

Еще одним примером использования дискретизации может служить цветное представление значений функции в некоторой одномерной, двумерной или трехмерной области, т.е. цветомодуляция функции. Можно рассматривать цветомодуляцию как некий оператор отображения численного значения функции $f(x)$ в вектор \overline{RGB} , компонентами которого являются цветовые составляющие пикселя:

$$C(f(x)) = (R, G, B), \quad (12)$$

В каждом узле сетки, покрывающей область, вычисляется значение функции, далее к этому значению применяется оператор цветомодуляции, и пиксель, соответствующий узлу сетки, закрасивается полученным цветом.

Для построения оператора цветомодуляции достаточно определить верхнюю и нижнюю границы значений функции в представляемой области. Обозначим их как $f_{\text{в}}$ и $f_{\text{н}}$ соответственно. Каждый из компонентов вектора \overline{RGB} принимает значения в отрезке $[0; 1]$. Задача построения оператора цветомодуляции сводится к построению линейного отображения отрезка $[f_{\text{н}}; f_{\text{в}}]$ в отрезок $[0; 1]$:

$$\varphi(x) = 1 + \frac{x - f_{\text{в}}}{f_{\text{в}} - f_{\text{н}}}. \quad (13)$$

В качестве примера рассмотрим цветомодуляцию от синего цвета к красному, где чистый синий цвет $(0, 0, 1)$ означает наименьшее, а чистый красный $(1, 0, 0)$ — наибольшее значения функции:

$$C : \begin{cases} R(f(x)) = 1 + \frac{f(x) - f_{\text{в}}}{f_{\text{в}} - f_{\text{н}}} \\ G(f(x)) = 0 \\ B(f(x)) = 1 + \frac{f(x) - f_{\text{н}}}{f_{\text{н}} - f_{\text{в}}} \end{cases}. \quad (14)$$

Цветомодуляция дает возможность визуально охарактеризовать поведение функции в области — увидеть направления и скорость ее возрастания и убывания, зоны ее экстремальных значений и т.п. Эффективным оказалось применение данного подхода к визуализации механической трактовки кратных интегралов. Например, визуализация двойного интеграла как массы тонкой пластинки с неравномерно распределенной плотностью, при этом значение подынтегральной функции представлялось с помощью цветомодуляции.

В заключение следует отметить, что перечисленные случаи использования известных численных методов в решении математических задач применительно к реализации обучающих программных продуктов по математике являются обобщением существующих разработок, апробированных в учебном процессе и подтвердивших эффективность их использования для упрощения, наглядности и доступности изучаемого материала.

ЛИТЕРАТУРА

- [1] *Возженников А.П.* Разработка тестирующего программного обеспечения на базе автоматического решения задач высшей математики // XVI конференция-выставка «Информационные технологии в образовании»: Сборник трудов участников конференции. Часть III. — М.: БИТ про, 2006. — С. 21—23.
- [2] *Соболь И.М.* Метод Монте-Карло. — М.: Наука, 1968.
- [3] *Бергман Г.Н.* Сборник задач по курсу математического анализа. — М.: Наука, 1985.

ABOUT SOME PARTICULAR CASES OF USING NUMERICAL METHODS IN SOLVING MATHEMATICAL PROBLEMS TO BE APPLIED IN TEACHING SOFTWARE DEVELOPMENT

A.P. Vozzhennikov

Department of High Mathematics and Programming Software
Murmansk State Technical University
Sportivnaya str., 13, Murmansk, Russia, 183010

Some particular cases of using of numerical methods in teaching software development are considered in the article: modification or compilation of known methods, attraction of statistical methods as an alternative of analytic formulas, digitization of the function's meanings; examples of using are cited, algorithms and mathematical models are described.