Решение методом Галеркина неоднородных краевых задач в Maxima

ревизия*№0

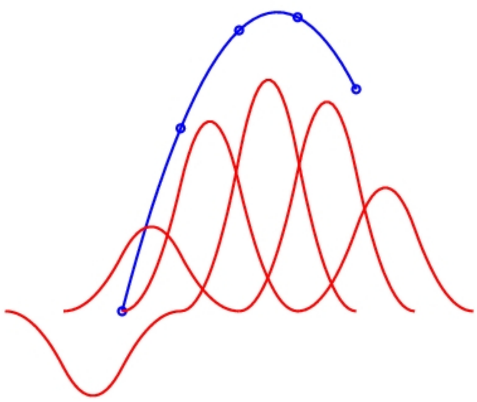
Дмитрий Мальков†

г. Комсомольск-на-Амуре

8 ноября 2014 г.

Аннотация

Продолжается знакомство с методом Галеркина. Приводится алгоритм решения неоднородных краевых задач и его реализация в системе символьной математики `Maxima`.



*статьи время от времени пересматриваются и исправляются, скачивайте свежайшие версии с домашней страницы автора

†**Email:** `maldmitrix@gmail.com`, **домашняя страница:** `https://maldmitrix.github.com`

Содержание

1 Определения краевой задачи и краевых условий **3**

2 Алгоритм решения обобщенным методом Галеркина **3**

3 Построение систем базисных и весовых функций **6**

4 Пример вычисления аппроксимации по алгоритму **8**

Литература **11**

Приложение А **12**

1 Определения краевой задачи и краевых условий

Будем рассматривать только краевые задачи для линейного обыкновенного дифференциального уравнения второго порядка.

Краевой задачей для функции $u(x)$ обобщенно называется задача вида

$$\begin{cases} \phi_1 \dfrac{d^2u}{dx^2} + \phi_2 \dfrac{du}{dx} + \phi_3 u = \phi_4, \quad a \le x \le b, \\ \left(\alpha u + \beta \dfrac{du}{dx}\right)\Bigg|_{x=a} = \omega_a, \quad \left(\gamma u + \delta \dfrac{du}{dx}\right)\Bigg|_{x=b} = \omega_b, \end{cases}$$

где $\phi_1, \phi_2, \phi_3, \phi_4$ – известные непрерывные на интервале $[a, b]$ функции от x; $\alpha, \beta, \gamma, \delta$ – известные коэффициенты; ω_a, ω_b – известные значения. Если первое уравнение разделить на $\phi_1 \neq 0$ и после этого переобозначить функции ϕ_i, то их станет всего три (как будет сделано чуть ниже). Символом L обозначается оператор линейного дифференциального уравнения. Первое уравнение в поставленной задаче можно коротко записать как $Lu = \phi_4$. Функция ϕ_4 не включается в состав оператора L, чтобы не нарушать его линейность относительно u.

Второе и третье уравнения называются *краевыми условиями*. Если $\omega_a = \omega_b = 0$, то такие краевые условия называются *однородными*.

В отличие от задачи Коши, краевая задача не всегда разрешима, а если разрешима, то не обязательно единственным образом.

2 Алгоритм решения обобщенным методом Галеркина

Пусть нужно решить задачу

$$\begin{cases} \dfrac{d^2u}{dx^2} + \phi_1 \dfrac{du}{dx} + \phi_2 u = \phi_3, \quad a \le x \le b, \\ \left(\alpha u + \beta \dfrac{du}{dx}\right)\Bigg|_{x=a} = \omega_a, \quad \left(\gamma u + \delta \dfrac{du}{dx}\right)\Bigg|_{x=b} = \omega_b, \end{cases}$$

причем в паре коэффициентов α и β есть хотя бы одно ненулевое, и то же самое относится к паре коэффициентов γ и δ (эту фразу можно математически записать как $\alpha^2 + \beta^2 > 0$ и $\gamma^2 + \delta^2 > 0$).

Везде далее будем предполагать существование единственного решения $u(x)$ для данной задачи, что часто вытекает из физического смысла того явления или процесса, математическое моделирование которого привело к задаче.

Зададимся на отрезке $[a,b]$ некоторой системой дважды непрерывно дифференцируемых функций $u_0(x), u_1(x), \ldots, u_n(x)$, таких, что $u_0(x)$ удовлетворяет данным в условии задачи краевым условиям, а остальные функции $u_1(x), u_2(x), \ldots, u_n(x)$, называемые базисными функциями, линейно независимы на отрезке $[a,b]$ и удовлетворяют однородным краевым условиям

$$\left(\alpha u + \beta \frac{du}{dx}\right)\Bigg|_{x=a} = 0, \quad \left(\gamma u + \delta \frac{du}{dx}\right)\Bigg|_{x=b} = 0.$$

Составим пробное решение задачи

$$u(x) \approx \tilde{u}(x) = u_0(x) + \sum_i c_i u_i(x), \quad i = 1, 2, \ldots, n,$$

с неизвестными пока коэффициентами $c_1, c_2, \ldots, c_n$. В силу линейности исходных краевых условий, пробное решение удовлетворяет им при любых значениях коэффициентов c_i. Подставляя пробное решение $\tilde{u}(x)$ в дифференциальное уравнение, находим выражение невязки:

$$R(c_i, x) = L\tilde{u} - \phi_3(x) = Lu_0 - \phi_3(x) + \sum_i c_i L u_i, \quad i = 1, 2, \ldots, n.$$

Невязка линейно зависит от параметров c_i и является характеристикой уклонения пробного решения $\tilde{u}(x)$ от точного решения $u(x)$. Если при некоторых значениях c_i невязка на рассматриваемом отрезке $[a,b]$ во всех точках равна нулю, то $u(x) \equiv \tilde{u}(x)$ в силу единственности точного решения (то есть мы случайно угадали точное решение). Но обычно невязка нулю не равна. Поэтому подбираем коэффициенты c_i так, чтобы невязка была в каком-то смысле наименьшей. В *обобщенном* методе Галеркина значения этих коэффициентов определяются из системы n уравнений

$$\langle\, R(c_1, c_2, \ldots, c_n, x), w_j(x) \,\rangle = \int_a^b R(c_1, c_2, \ldots, c_n, x) w_j(x)\, dx = 0,$$
$$j = 1, 2, \ldots, n,$$

где $w_j(x)$ – линейно независимые и непрерывные на интервале $[a,b]$ весовые функции (еще их называют поверочными). Если в качестве весовых функций w_j взять базисные функции u_j, то мы получаем *авторский* метод Галеркина. Если весовые функции w_j входят в полную систему функций, то при $n \to \infty$ невязка будет ортогональной всем элементам полной системы и, значит, сходится при $n \to \infty$ к нулю в среднем.

Запишем систему n уравнений:

$$\int_a^b \left[Lu_0 - \phi_3(x) + \sum_i c_i Lu_i \right] w_j(x)\,dx = 0, \quad i,j = 1,2,\ldots,n;$$

или

$$\sum_i \int_a^b c_i L u_i w_j(x)\,dx = \int_a^b \left[\phi_3(x) - Lu_0\right] w_j(x)\,dx, \quad i,j = 1,2,\ldots,n;$$

Перепишем систему развернутом виде:

$$\sum_i K_{ji} c_i = f_j, \quad i,j = 1,2,\ldots,n$$

или в матричной форме:

$$\mathbf{Kc} = \mathbf{f},$$

где $K_{ji} = \langle Lu_i, w_j(x)\rangle$, $f_j = \langle \phi_3(x) - Lu_0, w_j(x)\rangle = \langle -R(x), w_j(x)\rangle$. Решив эту систему (матричное уравнение), подставляем найденные коэффициенты c_i в разложение пробного решения $\tilde{u}(x)$ по базисным функциям и записываем ответ.

Запишем алгоритм решения неоднородной краевой задачи обобщенным методом Галеркина, предполагая, что пробное решение $\tilde{u}(x)$ задачи сходится к точному решению $u(x)$ при $n \to \infty$:

0. Выбираем главный член $u_0(x)$ пробного решения, базисные функции $u_1(x), u_2(x), \ldots, u_n(x)$ и весовые функции $w_1(x), w_2(x), \ldots, w_n(x)$.

   Находим выражение невязки при $\tilde{u}(x) = u_0(x)$: $R_0(x) = Lu_0(x) - \phi_3(x)$. Если для любого $x \in [a,b]$ невязка $R_0(x)$ равна нулю, то записываем ответ $u(x) \equiv u_0(x)$ и радуемся. В противном случае, переходим к следующему шагу алгоритма.

1. Берем только одну базисную функцию $u_1(x)$ и строим пробное решение первого шага в виде $\tilde{u}_{(1)}(x) = u_0(x) + c_1 u_1(x)$. Из описанной выше системы линейных уравнений при $n = 1$ (она выродится в одно уравнение) находим коэффициент c_1.

   Вычисляем невязку $R(c_1, x)$. Если для любого $x \in [a,b]$ невязка $R(c_1, x)$ равна нулю, то записываем ответ $u(x) \equiv u_0(x) + c_1 u_1(x)$ и заканчиваем вычисления. В противном случае, вычисляем то, как сильно пробное решение отклоняется от точного.

   Введем *меру точности* Δ_m пробного решения $\tilde{u}_{(m)}(x)$ на шаге m

(когда используются m различных базисных функций u_i):

$$\Delta_m = \max_{x\in[a,b]} |R(c_1, c_2, \dots, c_n, x)|, \quad i = 1, 2, \dots, m.$$

На первом шаге $m = 1$. Пусть нам нужно наибольшее отклонение, не превышающее некоторого заданного значения ε. Если $\Delta_m \le \varepsilon$, записываем ответ $u(x) \approx u_0(x)+c_1u_1(x)$ и заканчиваем вычисления. Иначе, переходим к следующему, $(m + 1)$-му, шагу. И так далее, пока не случится при некотором m, что $R(c_1, c_2, \dots, c_n, x) = 0$ или $\Delta_m \le \varepsilon$.

3 Построение систем базисных и весовых функций

Система функции называется *линейно независимой*, если любая линейная комбинация этих функций обращается в ноль тогда и только тогда, когда все коэффициенты линейной комбинации равны нулю.

Известно, что степенные функции $1, x, x^2, \dots, x^n, \dots$ линейно независимы на всей оси вещественных чисел $\mathbf{R}$ и, следовательно, на любом ее отрезке $[a, b] \subset \mathbf{R}$. Также на любом таком отрезке линейно независима любая система из многочленов *последовательных* степеней (то есть наивысшая степень переменной x у каждого многочлена системы уникальна).

Для построения $u_0(x)$ и линейно независимой на $[a, b]$ системы базисных функций $u_1(x), u_2(x), \dots, u_n(x)$, являющихся многочленами, можно применить метод *неопределенных коэффициентов*.

Пример 1. Построить главный член $u_0(x)$ и систему из пяти базисных функций $u_i(x),\ i \ge 1$ для задачи с краевыми условиями

$$\left(u(x) + \frac{du}{dx}\right)\Bigg|_{x=0} = 1, \quad \left(u(x) + \frac{du}{dx}\right)\Bigg|_{x=1} = -4.$$

Решение. Пусть главный член $u_0(x) = A$, тогда $\frac{du_0}{dx} = 0$, и краевые условия дают несовместную систему из уравнений $A = 1$ и $A = -4$.

Пусть теперь $u_0 = A + Bx$, тогда $\frac{du_0}{dx} = B$ и краевые условия дают

$$\begin{cases} A + B = 1, \\ A + 2B = -4; \end{cases} \quad\Leftrightarrow\quad \begin{cases} A + B = 1, \\ B = -5; \end{cases} \quad\Leftrightarrow\quad \begin{cases} A = 6, \\ B = -5. \end{cases}$$

Получаем $u_0 = 6 - 5x$.

Базисные функции $u_i(x),\ i \ge 1$ должны удовлетворять однородным условиям

$$\left(u(x)+\frac{du}{dx}\right)\Big|_{x=0}=0,\quad \left(u(x)+\frac{du}{dx}\right)\Big|_{x=1}=0.$$

Базисные функции не могут быть нулевыми, поэтому $u_1(x)$ в виде $u_1=A$ или $u_1=A+Bx$ не подойдет. Будем искать в виде $u_1=A+Bx+Cx^2$. Первая производная будет $\frac{du_1}{dx}=B+2Cx$, и из однородных условий получаем систему из двух уравнений для трех неизвестных (неопределенная система):

$$\begin{cases} A+B=0, \\ A+2B+3C=0; \end{cases} \quad \Leftrightarrow \quad \begin{cases} A+B=0, \\ B+3C=0 \end{cases}.$$

Имеем множество решений $A=3t, B=-3t, C=t$. Выбираем одно решение при $t=\frac{1}{3}$. Тогда

$$u_1(x)=1-x+\frac{1}{3}x^2.$$

Аналогично находим

$$u_2(x)=1-x+\frac{1}{4}x^3,\quad u_3(x)=1-x+\frac{1}{5}x^4,$$

$$u_4(x)=1-x+\frac{1}{6}x^5,\quad u_5(x)=1-x+\frac{1}{7}x^6.$$

В обобщенном методе Галеркина весовые функции w_j не обязательно должны быть такими же, как базисные. Можно выбрать, например, многочлены Лежандра, которые, как известно, ортогональны на интервале $t\in[-1,1]$:

$$P_n(t)=\frac{1}{2^n n!}\frac{d^n}{dt^n}(t^2-1)^n,\quad n=0,1,2,\ldots$$

Вместо параметра t нужно подставлять

$$t=\frac{2}{b-a}\left(x-\frac{a+b}{2}\right),$$

чтобы получить ортогональные на интервале $x\in[a,b]$ полиномы $P_n(x)$.

Дополнительные примеры и объяснения можно найти в неплохом учебном пособии А. В. Анкилова «Алгоритмы методов взвешенных невязок для решения линейных задач математической физики и их реализация в системе MathCAD», изданном в Ульяновском государственном техническом университете (2006 г.).

4 Пример вычисления аппроксимации по алгоритму

Рассмотрим подготовительный и первый шаги приведенного выше алгоритма для следующей краевой задачи:

$$\begin{cases} \dfrac{d^2u}{dx^2} - 3\dfrac{du}{dx} + 2u = 2x^2 - 6x + 2, \quad x \in [0,1], \\ \left(u + \dfrac{du}{dx}\right)\Bigg|_{x=0} = 1, \quad \left(u + \dfrac{du}{dx}\right)\Bigg|_{x=1} = -4. \end{cases}$$

Подготовительный шаг. Будем аппроксимировать степенными полиномами. Для данных краевых и однородных краевых условий мы уже находили функции u_0 и u_i $(i = 1, 2, \ldots, 5)$. Перепишем еще раз главный член разложения u_0:

$$u_0 = 6 - 5x.$$

Запишем для него выражение линейного дифференциального оператора:

$$Lu_0(x) = -3 \cdot (-5) + 2(6 - 5x) = 27 - 10x.$$

Запишем выражение невязки для главного члена:

$$R(x) = 27 - 10x - 2x^2 + 6x - 2 = -2x^2 - 4x + 25 \neq 0.$$

Значит, мы не угадали точное решение. Тогда переходим к следующему шагу. Но сначала найдем на всякий случай наибольшее отклонение невязки от нуля (по алгоритму это делать не обязательно). На отрезке $x \in [0,1]$ невязка плавно уменьшается со значения 25 до значения $25 - 2 - 4 = 19$. Значит $\Delta_0 = 25$.

Первый шаг. Представим пробное решение как

$$\tilde{u}_{(1)} = u_0 + c_1 u_1$$

где

$$u_0(x) = 6 - 5x, \quad u_1(x) = 1 - x + \frac{1}{3}x^2.$$

Находим выражение линейного дифференциального оператора для базисной функции u_1:

$$Lu_1(x) = \frac{2}{3} - 3\left(-1 + \frac{2}{3}x\right) + 2 - 2x + \frac{2}{3}x^2 = \frac{2}{3}x^2 - 4x + \frac{17}{3}.$$

Далее, не утруждая себя при интегрировании приведением подобных слагаемых, находим внутреннее произведение от функции u_1 и ее обработкой оператором $Lu_1(x)$:

$$K = \langle\ Lu_1(x), u_1(x)\ \rangle = \int_0^1 \left(\frac{2}{3}x^2 - 4x + \frac{17}{3}\right)\left(1 - x + \frac{1}{3}x^2\right)\, dx =$$
$$= \int_0^1 \left(\frac{2}{3}x^2 - 4x + \frac{17}{3} - \frac{2}{3}x^3 + 4x^2 - \frac{17}{3}x + \frac{2}{9}x^4 - \frac{4}{3}x^3 + \frac{17}{9}x^2\right)\, dx =$$
$$= \left(\frac{2}{9}x^3 - \frac{4}{2}x^2 + \frac{17}{3}x - \frac{2}{12}x^4 + \frac{4}{3}x^3 - \frac{17}{6}x^2 + \frac{2}{45}x^5 - \frac{4}{12}x^4 + \frac{17}{27}x^3\right)\Bigg|_0^1 =$$
$$= \frac{2}{9} - 2 + \frac{17}{3} - \frac{1}{6} + \frac{4}{3} - \frac{17}{6} + \frac{2}{45} - \frac{1}{3} + \frac{17}{27} \approx 2.563.$$

Таким же образом находим внутреннее произведение от функции u_1 и невязкой $R(x)$, полученной на шаге алгоритма $m = 0$ (при $\tilde{u}_{(0)} = u_0$) и взятой с обратным знаком:

$$f = \langle\ -R(x), u_1(x)\ \rangle = \int_0^1 \left(1 - x + \frac{1}{3}x^2\right)\left(2x^2 + 4x - 25\right)\, dx =$$
$$= \int_0^1 \left(2x^2 + 4x - 25 - 2x^3 - 4x^2 + 25x + \frac{2}{3}x^4 + \frac{4}{3}x^3 - \frac{25}{3}x^2\right)\, dx =$$
$$= \left(\frac{2}{3}x^3 + 2x^2 - 25x - \frac{2}{4}x^4 - \frac{4}{3}x^3 + \frac{25}{2}x^2 + \frac{2}{15}x^5 + \frac{4}{12}x^4 - \frac{25}{9}x^3\right)\Bigg|_0^1 =$$
$$= \frac{2}{3} + 2 - 25 - \frac{1}{2} - \frac{4}{3} + \frac{25}{2} + \frac{2}{15} + \frac{1}{3} - \frac{25}{9} \approx -13.978.$$

Из уравнения $Kc_1 = f$ находим нужный коэффициент:

$$c_1 = \frac{-13.978}{2.563} \approx -5.454.$$

Запишем пробное решение $\tilde{u}$ с найденным коэффициентом:

$$\tilde{u}_{(1)} = 6 - 5x + c_1(1 - x + \frac{1}{3}x^2) = \frac{1}{3}c_1x^2 - (5 + c_1)x + 6 + c_1.$$

Запишем выражение невязки от пробной функции на шаге $m = 1$ алгоритма:

$$R(c_1, x) = \frac{2}{3}c_1 - 2c_1x + 3c_1 + 15 + \frac{2}{3}c_1x^2 - 2c_1x - 10x + 12 + 2c_1 - 2x^2 + 6x - 2 =$$
$$= \left(\frac{2c_1 - 6}{3}\right)x^2 - 4(c_1 + 1)x + \left(\frac{17c_1}{3} + 25\right) \neq 0.$$

Точное решение снова не угадали. Найдем тогда наибольшее отклонение невязки $R(c_1, x)$ от нуля при первом шаге на отрезке $x \in [0, 1]$. Для этого нужно найти все локальные экстремумы функции $R(c_1, x)$, принадлежащие отрезку, найти в них значения невязки, а также найти значения невязки на концах отрезка, и выбрать из них наибольшее абсолютное значение.

Мы имеем параболу. Значит у нее есть только один экстремум, который может не принадлежать рассматриваемому отрезку. Для его нахождения, нужно взять первую производную и приравнять ее к нулю:

$$\frac{dR(c_1, x)}{dx} = \frac{2}{3}(2c_1 - 6)x - 4(c_1 + 1) = 0.$$

Отсюда найдем абсциссу единственного экстремума:

$$x = \frac{6(c_1 + 1)}{2c_1 - 6} \approx 1.58 \notin [0; 1].$$

Экстремум находится вне рассматриваемого отрезка. Найдем значения невязки на концах отрезка:

$$R(c_1, 0) \approx -5.906, \quad R(c_1, 1) \approx 6.274.$$

Выбираем наибольшее абсолютное значение:

$$\Delta_1 = \max_{x \in [0,1]} |R(c_1, x)| \approx 6.274.$$

Теперь нужно оценить погрешность. Если бы область задачи была, скажем, $[0, 50\,000]$, то наибольшая погрешность, равная в некоторой точке 6.274, была бы еще приемлемой. Но на маленьком интервале $[0, 1]$ полученная аппроксимация не может серьезно рассматриваться. Введем ограничение на приемлемость погрешности: $\varepsilon = 0.01$. Алгоритм теперь нужно выполнять до тех пор, пока Δ_m не станет при некотором шаге m меньше предела неприемлемости аппроксимации ε.

Алгоритм выполнен до четвертого шага включительно в программе `Maxima`. Реализацию смотрите в приложении А. Поставленная задача имеет единственное точное решение вида

$$u(x) = \frac{e^2 + 7}{2(e^2 - e)} e^x - \frac{7 + e}{3(e^2 - e)} e^{2x} + x^2.$$

Список литературы

[1] Анкилов, А. В. Алгоритмы методов взвешенных невязок для решения линейных задач математической физики и их реализация в системе MathCAD: учебное пособие / А. В. Анкилов, П. А. Вельмисов, А. С. Семёнов. – Ульяновск : УлГТУ, 2006. – 168 с.

[2] Флетчер, К. Численные методы на основе метода Галеркина / К. Флетчер.– М. : Мир, 1988. – 352 с.

[3] Bruce A. Finlayson The method of weighted residuals and variational principles. – Academic ress, inc., New York, 1972.

[4] Иванов В.Н. Вариационные принципы и методы решения задач теории упругости: Учеб. пособие – М.: Изд-во РУДН, 2004. – 176 с.: ил.

[5] Бабенко К.И. Основы численного анализа. – М.: Наука, 1986.

Приложение А

Код для программы Maxima. Для его запуска нужно в консоли ввести `maxima -b` *Путь-К-Файлу*. Код вычерчивает три графика.

```
/************  файл "galerkin.mac" ******************/
/* запускать командой << maxima -b ./galerkin.mac >> */
/*****************  НАЧАЛО  ***********************/



/**************************
**  Подготовительный шаг  **
**************************/

/* Задание линейного дифференциального оператора,
   функции для правой части уравнения,
   общего выражения невязки */
L(y,x)  := diff(y,x,2) - 3*diff(y,x,1) + 2*y$
Pfi3(x) := 2*x^2 - 6*x + 2$
R(y,x)  := L(y,x) - Pfi3(x)$

/* Задание главного члена пробной функции
*  и базисных функций */
u0(x) := 6 - 5*x$
u1(x) := 1 - x + 1/3*x^2$
u2(x) := 1 - x + 1/4*x^3$
u3(x) := 1 - x + 1/5*x^4$
u4(x) := 1 - x + 1/6*x^5$

/* Невязка только от главной функции */
RU0(x) := R(u0(x),x)$

/* Левая часть дифф.уравнения при каждой из базисных функций */
Lu1(x) := L(u1(x),x)$
Lu2(x) := L(u2(x),x)$
Lu3(x) := L(u3(x),x)$
Lu4(x) := L(u4(x),x)$

/* Общее выражение внутреннего
*  произведения двух произвольных функций */
InProd(y1, y2,x,a,b) := integrate(y1(x)*y2(x),x,a,b)$



/****************
**  Шаг первый  **
****************/

/* Преобразованное условие ортогональности */
K1: InProd(Lu1,u1,x,0,1)$
```

```
f1: InProd(RU0,u1,x,0,1)*(-1)$

/* Ищем коэффициент */
c1: f1/K1$

/* Записываем пробное решение */
U1(x) := u0(x) + c1*u1(x)$

/* Невязка на первом шаге */
RU1(x) := R(U1(x),x)$

/****************
**  Шаг второй  **
****************/

/* Матричное уравнение ортогональности */
K2: matrix(  [InProd(Lu1,u1,x,0,1),  InProd(Lu2,u1,x,0,1)],
             [InProd(Lu1,u2,x,0,1),  InProd(Lu2,u2,x,0,1)])$

f2: matrix(  [InProd(RU0,u1,x,0,1)*(-1) ],
                  [InProd(RU0,u2,x,0,1)*(-1) ] )$

/* Ищем коэффициенты */
c2: invert(K2).f2$

/* Записываем пробное решение */
U2(x) := u0(x) + c2[1]*u1(x)+c2[2]*u2(x)$

/* Невязка на втором шаге */
RU2(x) := R(U2(x)[1],x)$



/****************
**  Шаг третий  **
****************/

/* Матричное уравнение ортогональности */
K3: matrix(
    [InProd(Lu1,u1,x,0,1),  InProd(Lu2,u1,x,0,1),
                 InProd(Lu3,u1,x,0,1) ],
    [InProd(Lu1,u2,x,0,1),  InProd(Lu2,u2,x,0,1),
                 InProd(Lu3,u2,x,0,1)] ,
    [InProd(Lu1,u3,x,0,1),  InProd(Lu2,u3,x,0,1),
                 InProd(Lu3,u3,x,0,1)] )$

f3: matrix(  [InProd(RU0,u1,x,0,1)*(-1) ],
             [InProd(RU0,u2,x,0,1)*(-1) ],
             [InProd(RU0,u3,x,0,1)*(-1) ] )$

/* Ищем коэффициенты */
c3:  invert(K3).f3$
```

```
/* Записываем пробное решение */
U3(x) := u0(x) + c3[1]*u1(x) + c3[2]*u2(x) + c3[3]*u3(x)$

/* Невязка на третьем шаге */
RU3(x) := R(U3(x)[1],x)$



/*******************
**  Шаг четвертый  **
*******************/

/* Матричное уравнение ортогональности */
K4: matrix(  [InProd(Lu1,u1,x,0,1), InProd(Lu2,u1,x,0,1),
              InProd(Lu3,u1,x,0,1), InProd(Lu4,u1,x,0,1)],
             [InProd(Lu1,u2,x,0,1), InProd(Lu2,u2,x,0,1),
              InProd(Lu3,u2,x,0,1), InProd(Lu4,u2,x,0,1)],
             [InProd(Lu1,u3,x,0,1), InProd(Lu2,u3,x,0,1),
              InProd(Lu3,u3,x,0,1), InProd(Lu4,u3,x,0,1)],
             [InProd(Lu1,u4,x,0,1), InProd(Lu2,u4,x,0,1),
              InProd(Lu3,u4,x,0,1), InProd(Lu4,u4,x,0,1)])$

f4: matrix(  [InProd(RU0,u1,x,0,1)*(-1)],
             [InProd(RU0,u2,x,0,1)*(-1)],
             [InProd(RU0,u3,x,0,1)*(-1)],
             [InProd(RU0,u4,x,0,1)*(-1)])$

/* Ищем коэффициенты */
c4:  invert(K4).f4$

/* Записываем пробное решение */
U4(x) := u0(x) + c4[1]*u1(x) + c4[2]*u2(x)
+ c4[3]*u3(x) + c4[4]*u4(x)$

/* Невязка на третьем шаге */
RU4(x) := R(U4(x)[1],x)$



/********************
**  Точное решение  **
********************/
Uexact(x) := 1/2*(%e^2 + 7)/(%e^2 - %e)*%e^x -
(7 + %e)/3/(%e^2 - %e)*%e^(2*x) + x^2$



/*************
**  Графики  **
*************/
```

```
/* Графики невязок */
plot2d( [ RU1(x), RU2(x), RU3(x), RU4(x) ], [x,0,1],
 [plot_format, gnuplot],[legend,"Шаг m = 1",
"Шаг m = 2","Шаг m = 3","Шаг m = 4"],
[ylabel, "Невязка R(x)"], [nticks,100])$

/* Графики пробных решений */
plot2d( [ U1(x), U2(x)[1], U3(x)[1], U4(x)[1]],
[x,0,1], [plot_format, gnuplot],
[legend,"Шаг m = 1", "Шаг m = 2", "Шаг m = 3", "Шаг m = 4"],
[ylabel, "Пробное решение U(x)"], [nticks,100])$

/* График точного решения и последнее пробное решение */
plot2d( [ U4(x)[1], Uexact(x)], [x,0,1], [plot_format, gnuplot],
[legend,"Шаг m = 4", "Точное реш."],
[ylabel, "Пробное решение U(x)"], [nticks,100])$

/*****************  КОНЕЦ  ******************/
```

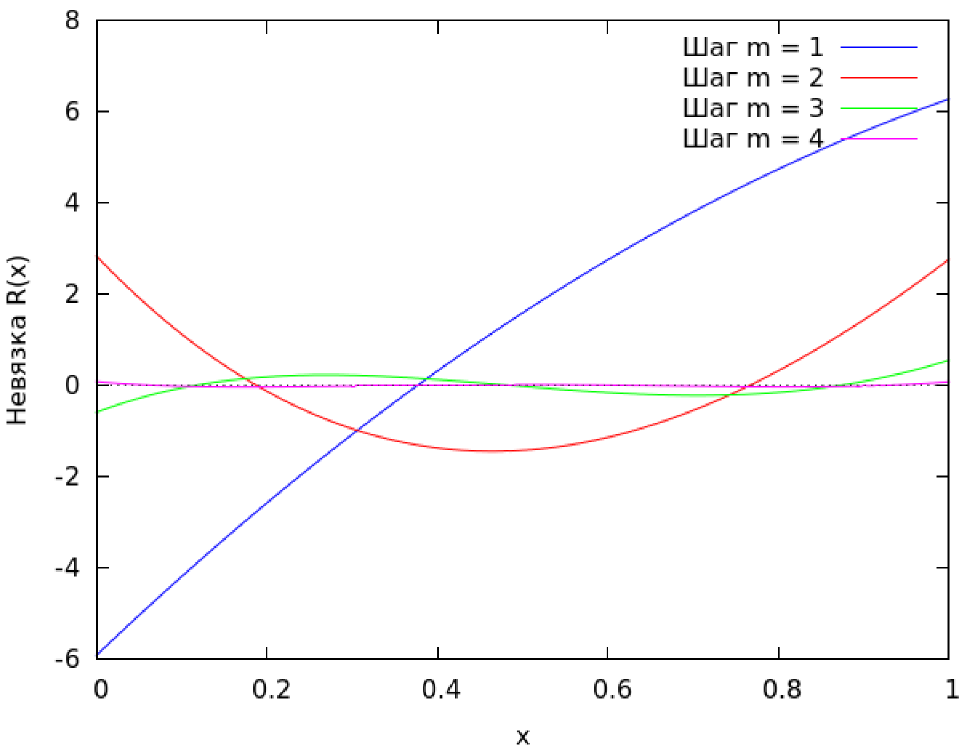
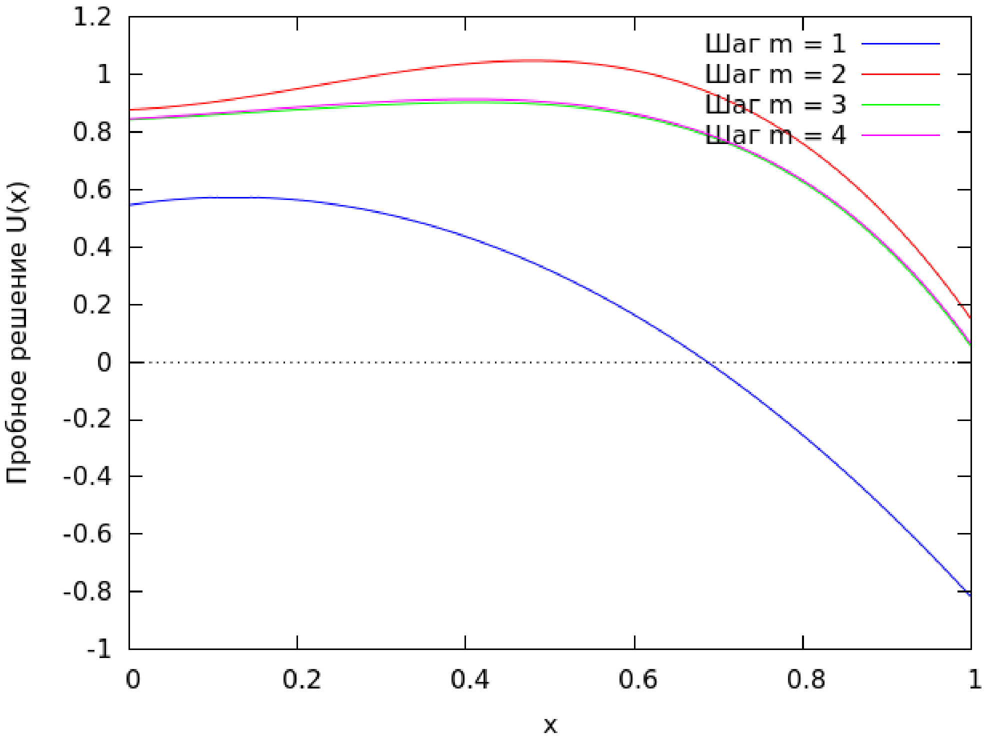
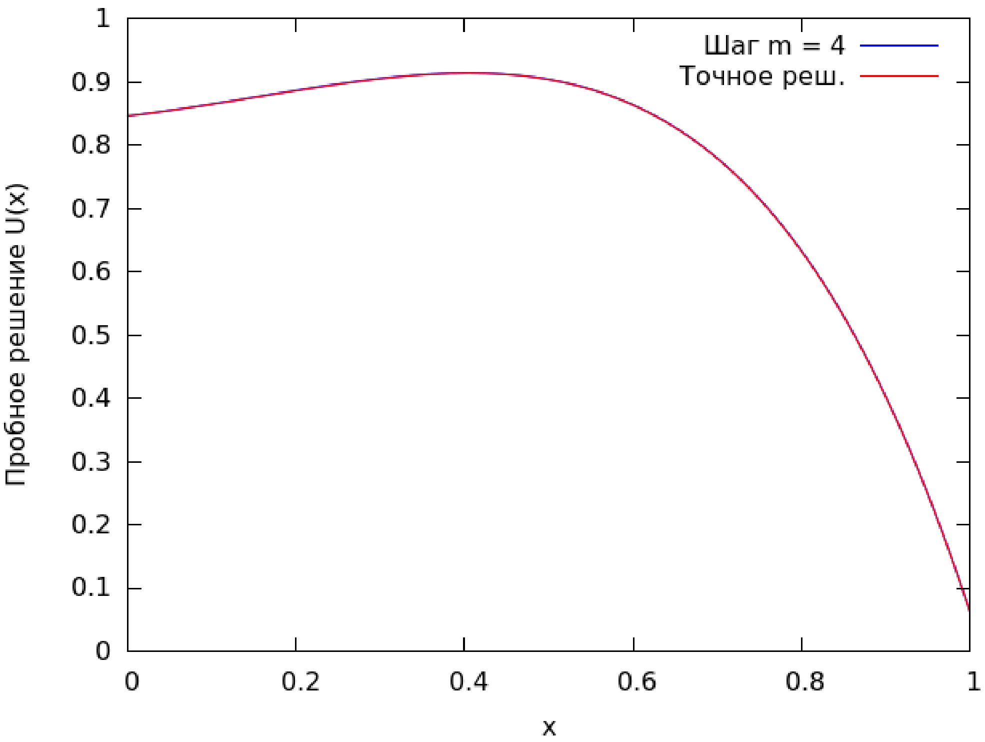


Рис. 1: Графики невязок, полученные на первых четырех шагах алгоритма.



а)



б)

Рис. 2: а) Графики пробных решений, полученных на первых четырех шагах алгоритма. б) Графики точного и четвертого пробного решений