МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М. В. ЛОМОНОСОВА

Научно-исследовательский вычислительный центр

ПРАКТИКУМ НА ЭВМ

ПО ВЫЧИСЛИТЕЛЬНЫМ МЕТОДАМ

О. Б. Арушанян

ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Работа поддержана РФФИ (проект 01–07–90173)

Москва, 2003

О. Б. Арушанян

З А Д А Н И Е

вычислительного практикума на ЭВМ

“Численное решение нелинейных уравнений”

1. Цель задания

1) Практика в использовании ЭВМ в дистанционном режиме.

2) Практика в использовании численных методов решения нелинейных уравнений.

3) Практика в использовании стандартных библиотечных программ.

4) Практика в модульном программировании.

2. Постановка задачи

Пусть дано уравнение

$$f(x) = 0, \tag{1}$$

где функция $f(x)$ определена и непрерывна на заданном конечном или бесконечном отрезке $[A, B]$. Всякое значение $\bar{x}$, обращающее функцию $f(x)$ в нуль, т.е. такое, что $f(\bar{x}) = 0$, называется *корнем* уравнения (1) или *нулем* функции $f(x)$.

Постановка 1. Найти все корни уравнения (1).

Очевидно, что такая постановка задачи неконструктивна, поскольку она никак не касается вопроса о существовании корней. Более правомерной является следующая

Постановка 2. Корни уравнения (1) существуют. Найти их.

Но в этом случае даже простой пример уравнения $\operatorname{tg} x - x = 0$ показывает принципиальную невозможность решить задачу в данной постановке, поскольку вычисление бесконечного числа корней требует бесконечно много времени. Естественно, в этой связи на первый план выдвигается вопрос об *отделении* (локализации) корней, т.е. об определении таких отрезков $[a_i, b_i] \in [A, B]$, каждый из которых содержит единственный корень уравнения (1). Рассмотрим вкратце наиболее распространенные способы отделения корней.

Если непрерывная функция $f(x)$ принимает значения разных знаков на концах отрезка $[\alpha, \beta] \in [A, B]$, т.е. $f(\alpha)f(\beta) < 0$, то внутри этого отрезка лежит нечетное число корней уравнения $f(x) = 0$ (по меньшей мере один). Корень заведомо будет единственным, или *простым*, если производная $f'(x)$ существует и сохраняет постоянный знак внутри интервала (α, β), т.е. если $f'(x) > 0$ (или $f'(x) < 0$) при $\alpha < x < \beta$.

На рис. 1 изображена функция $y = f(x)$, принимающая на концах отрезка $[\alpha, \beta]$ разные знаки. Поскольку $f'(x)$ на этом отрезке не является знакопостоянной, то внутри него лежат несколько корней уравнения $f(x) = 0$. На отрезке $[\alpha, \beta_1]$ имеем $f'(x) < 0$, и поэтому корень $\bar{x}_1$, принадлежащий этому отрезку, является простым. Корень $\bar{x}_2$ имеет четную кратность.

Табличный способ, легко поддающийся программной реализации, состоит в определении знаков функции $f(x)$ в узлах некоторой сетки $\{\alpha_1 = A, \alpha_2, \ldots, \alpha_{k-1}, \alpha_k = B\}$, выбор которой делается с учетом особенностей $f(x)$. Если в полученной таблице значений $f(x)$ имеем $f(\alpha_i)f(\alpha_{i+1}) < 0$, то на отрезке $[\alpha_i, \alpha_{i+1}]$ имеется хотя бы один корень уравнения $f(x) = 0$.

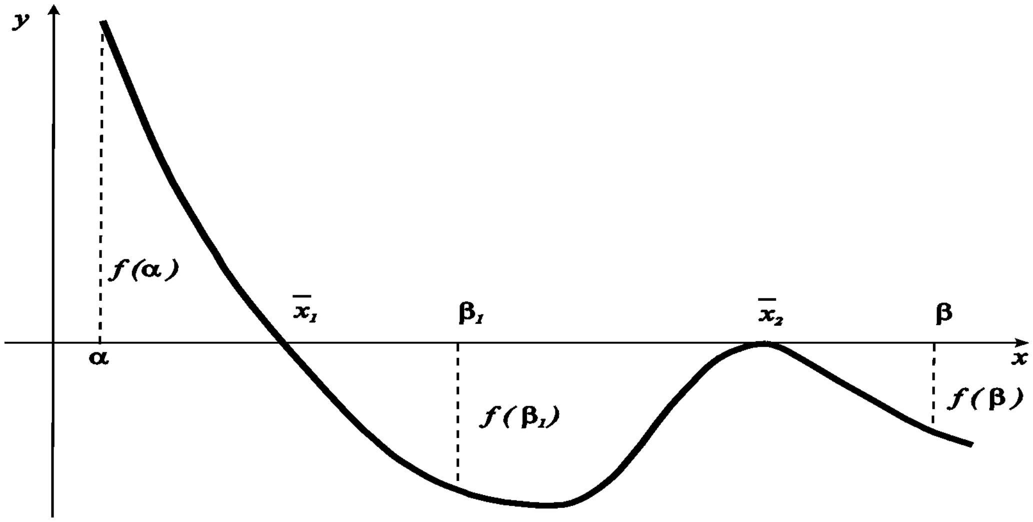


Рис. 1.

Если эти узлы близки, то, скорее всего, корень между ними один. Однако в этом нужно убедиться тем или иным способом. Выявить по таблице корни четной кратности сложно.

Если существует непрерывная производная $f'(x)$ и корни уравнения

$$f'(x) = 0$$

легко вычисляются, то процесс отделения корней уравнения (1) можно упорядочить. Для этого достаточно определить знаки функции $f(x)$ в точках корней ее производной и в граничных точках $x = A$ и $x = B$.

Более наглядным способом, который дает неплохие приближенные значения корней, является *графический* способ. По полученной таблице значений функции $f(x)$ строится ее график и графически ищутся точки пересечения его с осью абсцисс. Во многих задачах техники такая точность уже достаточна. Построение графика зачастую помогает также выявить корни четной кратности. Графический способ эффективен, если уравнение (1) не имеет близких между собой корней.

На практике часто удобно применять прием, состоящий в замене уравнения (1) эквивалентным ему уравнением $\varphi(x) = \psi(x)$, в котором функции $y_1 = \varphi(x)$ и $y_2 = \psi(x)$ имеют более простые графики. Абсциссы точек пересечения этих графиков будут корнями исходного уравнения. Например, уравнение $x \sin x - 1 = 0$ удобно преобразовать к виду $\sin x = 1/x$.

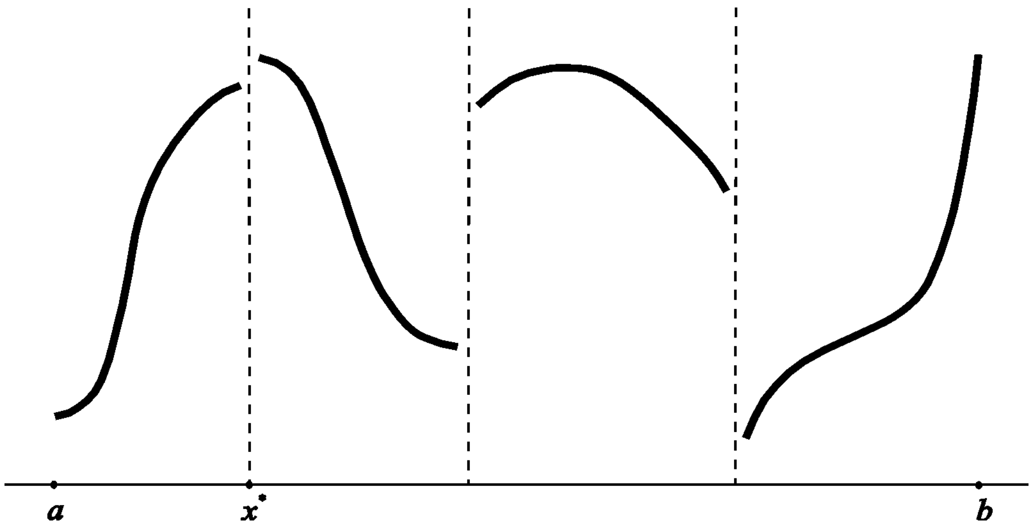
Хотя графический способ удобен и сравнительно прост, он применим, как правило, лишь для грубого определения корней. Особенно значительная потеря точности имеет место в тех случаях, когда линии пересекаются под очень острым углом и практически сливаются по некоторой дуге.

После решения вопроса об отделении корней становится правомерной следующая

Постановка 3. Уравнение (1) имеет на отрезке $[a, b]$ единственный корень. Найти его.

Это уже лучше поставленная задача, но еще очень сложная. Убедиться в этом позволяет следующий рисунок:

Если к тому же искомый корень x^* является иррациональным, то данная задача становится принципиально неразрешимой, поскольку иррациональные числа на ЭВМ представлять невозможно. Тогда в этом случае постановку задачи следует дополнить тем, что необходимо



найти не сам иррациональный корень, а "хорошее" рациональное приближение к нему, причем такое, которое может поместиться в разрядной сетке машинного слова. Однако если это возможно, то функция $f(x)$ должна быть непрерывной на отрезке $[a, b]$. В результате мы пришли еще к одной постановке задачи, свободной от рассмотренных недостатков:

Постановка 4. Пусть уравнение (1) имеет на отрезке $[a, b]$ единственный корень x^* и функция $f(x)$ непрерывна на $[a, b]$. Найти этот корень.

Из данной постановки следует, что $f(x) = (x - x^*)^n g(x)$, где функция $g(x)$ сохраняет знак на $[a, b]$. Это представление $f(x)$ явно указывает на два возможных качественных различия в ее поведении в окрестности корня: при нечетном n функция $f(x)$ меняет знак на $[a, b]$, а при четном n — нет.

Только после того как известны ответы на вопросы о существовании, единственности и кратности корня на $[a, b]$, следует переходить к выбору подходящего численного метода.

3. Численные методы решения нелинейного уравнения

В настоящем разделе дается краткое описание тех численных методов решения нелинейного уравнения, применение которых предусмотрено в данном задании. Здесь предполагается, что корень уравнения (1) уже отделен, т.е. $f(A)f(B) < 0$.

3.1. Метод половинного деления

Данный метод называют еще методом деления пополам, дихотомией, методом бисекций, или методом вилки. Положим $x_0 = A$ и $x_1 = B$. Найдем середину отрезка $x_2 = (x_0 + x_1)/2$, вычислим $f(x_2)$ и из полученных двух отрезков выберем тот, на концах которого функция имеет разные знаки, поскольку один из корней лежит на этом отрезке. Затем выбранный отрезок делим пополам и снова выбираем из полученных двух отрезков тот, на концах которого функция имеет разные знаки, и т.д. Если требуется найти корень с абсолютной точностью ε, то продолжаем деление пополам до тех пор, пока длина отрезка не станет меньше 2ε. Тогда середина последнего отрезка даст значение корня с требуемой точностью. Геометрическая интерпретация метода иллюстрируется на рис. 2.

В результате итерационного применения описанной процедуры на N этапе мы получаем или точный корень уравнения (1), или же бесконечную последовательность вложенных друг в друга отрезков, содержащих искомый корень, причем длина самого внутреннего отрезка уменьшилась по сравнению с длиной исходного отрезка в 2^N раз.

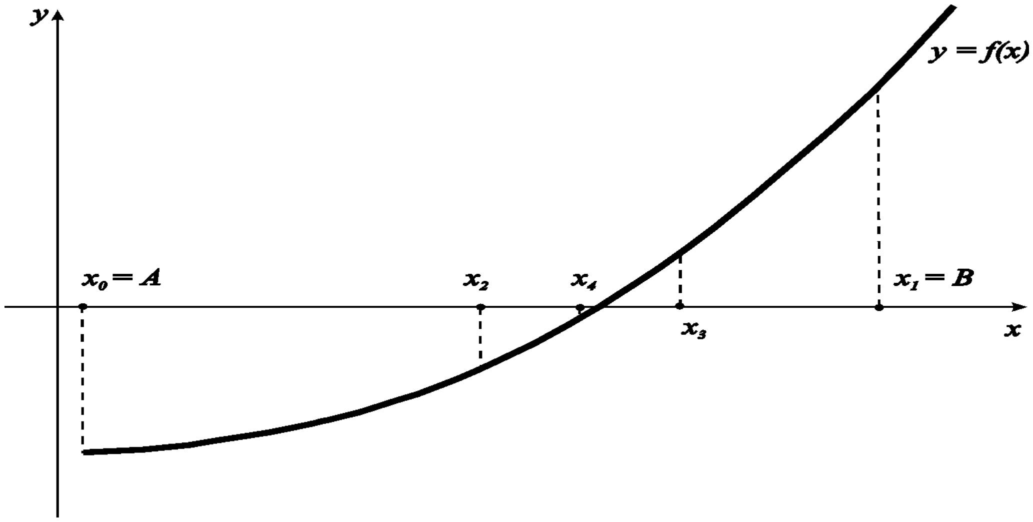


Рис. 2.

3.2. Метод простых итераций

Уравнение (1) можно заменить эквивалентным ему уравнением

$$x = \varphi(x). \tag{2}$$

Такую замену можно сделать многими способами, например, положив

$$\begin{aligned} \varphi(x) &= x + \alpha f(x), \quad \text{где} \quad \alpha = \text{const}, \\ \varphi(x) &= x + \psi(x) f(x), \end{aligned}$$

где $\psi(x)$ — произвольная непрерывная знакопостоянная функция, т.е. не имеющая нулей на отрезке $[A, B]$.

Выберем некоторое нулевое приближение корня $x_0 \in [A, B]$ уравнения (2), а дальнейшие приближения будем вычислять по формулам

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \ldots . \tag{3}$$

Очевидно, что если последовательность x_n стремится к некоторому пределу $\bar{x}$, то этот предел и есть искомый корень уравнения (2). Это будет иметь место в том случае, когда отображение $y = \varphi(x)$ при некотором $q < 1$ удовлетворяет условию

$$|\varphi(x_1) - \varphi(x_2)| \le q\,|x_1 - x_2|$$

для всех x_1 и x_2 из $[A, B]$, т.е. является *сжимающим.*

Дадим геометрическую интерпретацию метода простых итераций. Для этого построим на плоскости xOy графики функций $y = x$ и $y = \varphi(x)$. Каждый вещественный корень $\bar{x}$ уравнения (2) является абсциссой точки пересечения M кривой $y = \varphi(x)$ с прямой $y = x$ (рис. 3). Начиная с некоторой точки $A_0[x_0; \varphi(x_0)]$ строим ломанные линии $A_0B_1A_1B_2A_2 \ldots$ ("лестница"), звенья которой попеременно параллельны оси Ox и оси Oy, причем вершины $A_0, A_1, A_2, \ldots$ лежат на кривой $y = \varphi(x)$. Общие абсциссы точек A_1 и B_1, A_2 и B_2, ... представляют собой последовательные приближения $x_1, x_2, \ldots$ корня $\bar{x}$, которые сходятся к нему *монотонно* и *односторонне.*

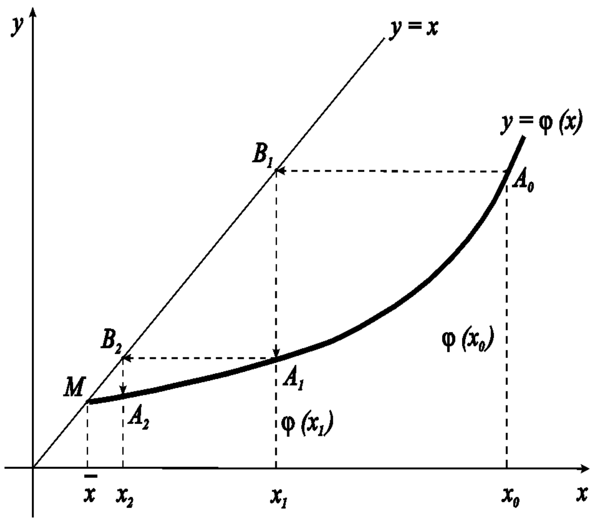


Рис. 3.

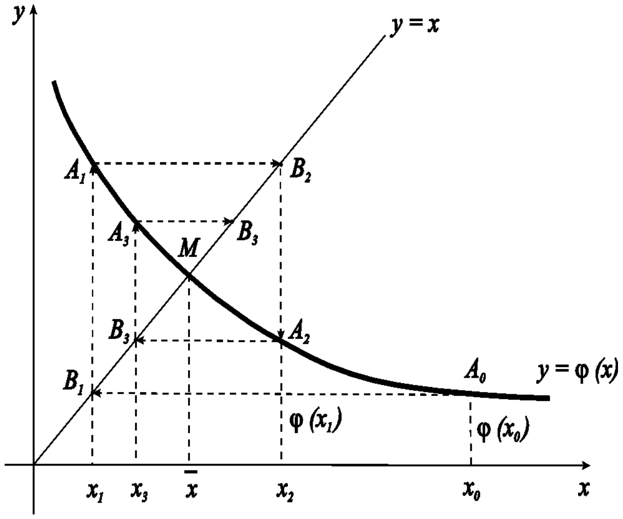


Рис. 4.

На рис. 3 изображен случай, когда $0 < \varphi'(x) < 1$. Если же $-1 < \varphi'(x) < 0$, то ломаная $A_0B_1A_1B_2A_2\ \ldots$ имеет вид "спирали" (рис. 4).

В этом случае сходимость к корню является *двухсторонней*, т.е. искомый корень $\bar{x}$ всегда принадлежит отрезку $[x_n, x_{n+1}]$.

3.3. Метод Ньютона

Данный метод называют методом касательных, или линеаризации. Расчетная формула метода Ньютона в некоторой окрестности корня уравнения (1) имеет вид (начальное приближение $x_0 \in [A, B]$):

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{4}$$

Его геометрическая интерпретация для различных случаев поведения функции $f(x)$ приведена на рис. 5. Из рис. 5 видно, что метод Ньютона состоит в замене на каждой итерации небольшой дуги кривой $y = f(x)$ касательной к ней. Это следует из уравнения касательной, проведенной к точке $(x_n, f(x_n))$:

$$y - f(x_n) = f'(x_n)(x - x_n),$$

из которой процесс (4) получается, если положить $y = 0$ и $x = x_{n+1}$. Из рис. 5 видно, что последовательность x_n монотонно и односторонне сходится к корню $\bar{x}$.

3.4. Комбинированный метод

Данный метод представляет собой комбинацию метода хорд и метода Ньютона. Дадим сначала основную идею метода хорд.

Сущность метода хорд видна из его геометрической интерпретации для различных случаев поведения $f(x)$ (рис. 6) и состоит в замене кривой $y = f(x)$ хордами, проходящими через концы отрезков, в которых $f(x)$ имеет противоположные знаки. Уравнение хорды, проходящей через точки с координатами $(A, f(A))$ и $(B, f(B))$, имеет вид

$$\frac{x - A}{B - A} = \frac{y - f(A)}{f(B) - f(A)}.$$

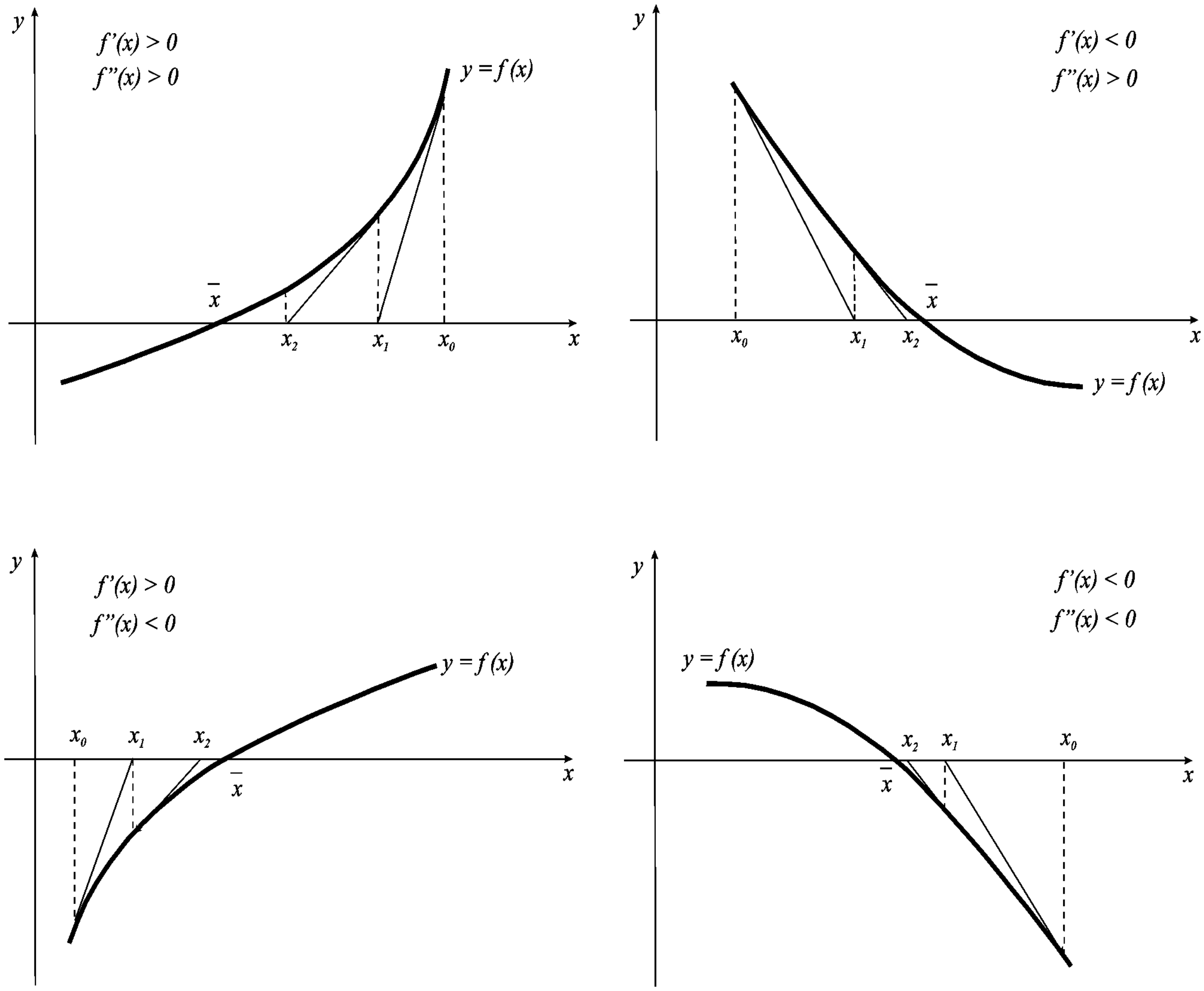


Рис. 5.

Исходя из этого уравнения строится итерационный процесс, сходящийся для рассмотренных случаев к корню $\bar{x}$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(x_0)}\,(x_n - x_0), \quad n = 1, 2, \ldots . \tag{5}$$

Видно, что этот метод является *двухшаговым* (т.е. для получения следующего приближения нужно знать значения $f(x)$ в двух точках) и требует, чтобы один конец отрезка, на котором ищется корень, был неподвижен. В качестве неподвижного конца выбирается тот конец, для которого знак $f(x)$ совпадает со знаком ее второй производной $f''(x)$. Тогда последовательные приближения x_n лежат по ту сторону корня, где $f(x)$ имеет знак, противоположный $f''(x)$. Сходимость метода хорд — односторонняя и монотонная. Так как на каждом шаге итерационного процесса за приближенное значение корня x_{n+1} принимается корень интерполяционного многочлена первой степени, то метод хорд называется еще методом *линейной интерполяции*.

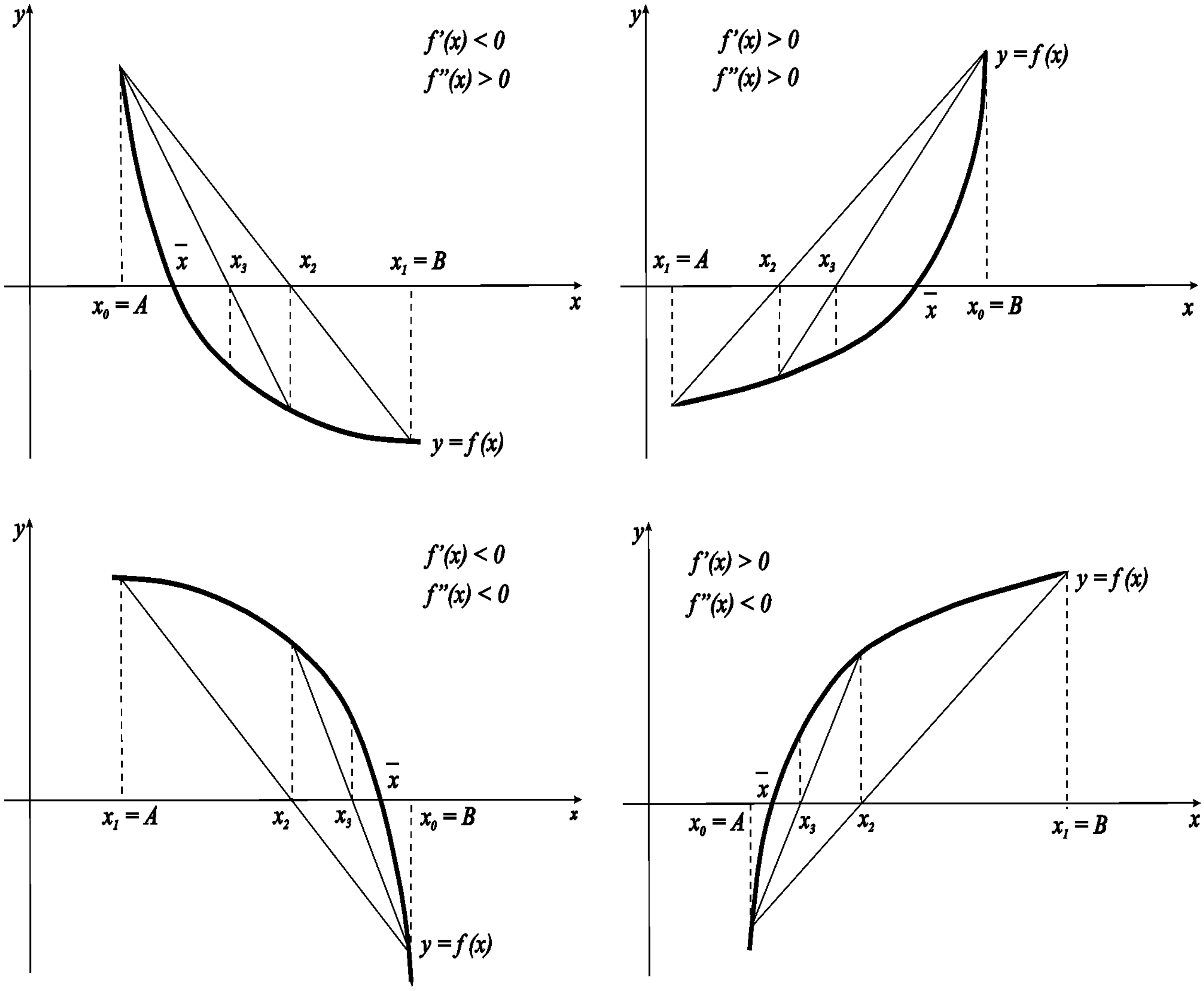


Рис. 6.

Комбинируя методы хорд и Ньютона, получаем следующие расчетные формулы:

$$
\begin{aligned}
\tilde{x}_{n+1} &= \tilde{x}_n - \frac{f(\tilde{x}_n)}{f'(x_n)}, \\
x_{n+1} &= x_n - \frac{f(x_n)}{f(\tilde{x}_n) - f(x_n)}\,(\tilde{x}_n - x_n), \quad n = 0, 1, 2, \ldots .
\end{aligned}
\tag{6}
$$

В этих формулах метод хорд применяется каждый раз к новому отрезку $[x_n, \tilde{x}_n]$. Геометрическая интерпретация комбинированного метода в случае $f'(x) > 0$, $f''(x) > 0$ на $[A, B]$ приведена на рис. 7. Для других случаев геометрическая интерпретация следует из рис. 6.

4. Скорость и условия сходимости методов, выбор начального приближения

Рассмотрим более строго изложенные методы, уделив внимание анализу скорости и условиям их сходимости, а также надлежащему выбору начального приближения к корню.

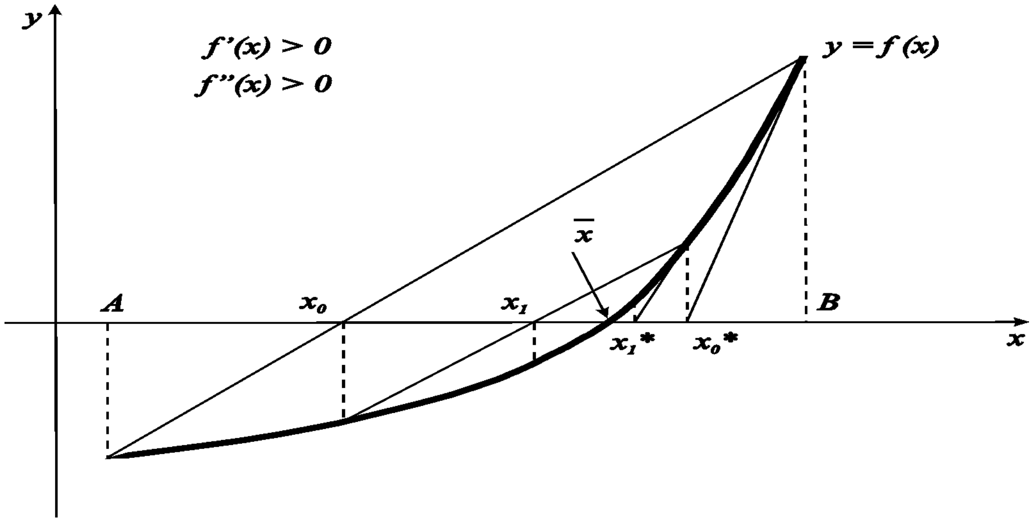


Рис. 7. $x_i^* = \tilde{x}_i$

4.1. Метод половинного деления

Для данного метода сходимость к простому корню обеспечивается для любых непрерывных функций $f(x)$ (в том числе недифференцируемых), для которых выполняется условие $f(A)f(B) < 0$. Метод сходится со скоростью геометрической прогрессии со знаменателем $1/2$, при этом выбирать начальное приближение к корню не нужно. За одну итерацию точность увеличивается вдвое, т.е. уточнение трех цифр требует 10 итераций.

4.2. Метод простой итерации

Исследуем условия сходимости метода. Если $\varphi(x)$ имеет непрерывную производную, то

$$x_{n+1} - \bar{x} = \varphi(x_n) - \varphi(\bar{x}) = (x_n - \bar{x})\,\varphi'(\xi),$$

где точка ξ лежит между точками x_n и $\bar{x}$. Поэтому если всюду на $[A, B]$ имеем $|\varphi'(x)| \le q < 1$, то отрезки $|x_n - \bar{x}|$ убывают не медленней членов геометрической прогрессии со знаменателем $q < 1$, и последовательность x_n сходится при *любом* нулевом приближении x_0.

Если $|\varphi'(x)| > 1$, то в силу непрерывности $|\varphi'(x)|$ больше единицы и в некоторой окрестности корня. В этом случае итерации сходиться не будут. Это иллюстрируется на рис. 8 для $\varphi'(x) > 1$ и $\varphi'(x) < -1$.

Если $|\varphi'(x)| < 1$ в некоторой окрестности корня, но вдали от него $|\varphi'(x)| > 1$, то итерации будут сходиться, когда нулевое приближение x_0 выбрано достаточно близко к корню. В этом случае при произвольном нулевом приближении сходимости может не быть.

Очевидно, что чем меньше q, тем быстрей сходимость. Вблизи корня асимптотическая сходимость определяется величиной $|\varphi'(\bar{x})|$ и будет особенно быстрой при $\varphi'(\bar{x}) = 0$. Следовательно, успех метода зависит от того, насколько удачно выбрана функция $\varphi(x)$. Например, для извлечения квадратного корня, т.е. решения уравнения $x^2 = a$, можно положить $\varphi(x) = a/x$ или $\varphi(x) = (x + a/x)/2$ и записать такие итерационные процессы

$$x_{n+1} = \frac{a}{x_n} \qquad \text{или} \qquad x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right).$$

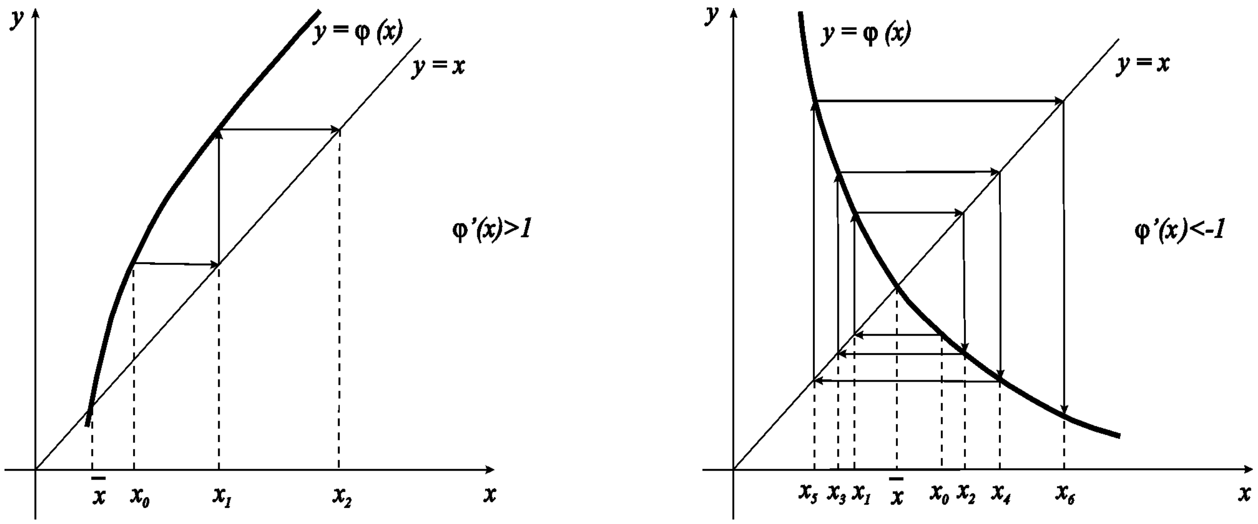


Рис. 8.

Первый процесс вообще не сходится, а второй сходится при любом x_0 и очень быстро, поскольку $\varphi'(\bar{x}) = 0$.

Погрешность $|x_n - \bar{x}|$ стремится к нулю монотонно при сходящемся итерационном процессе (3), т.к.

$$|x_n - \bar{x}| \le q|x_{n-1} - \bar{x}| \le |x_{n-1} - \bar{x}|.$$

Это означает, что каждое следующее значение x_n является (без учета ошибок округления) более точным, чем предшествующее значение x_{n-1}.

4.3. Метод Ньютона

Будем предполагать, что на отрезке $[A, B]$, $f(A)f(B) < 0$, в котором ищется корень уравнения (1), $f'(x)$ и $f''(x)$ непрерывны и сохраняют постоянные знаки. По формуле Лагранжа имеем

$$0 = f(\bar{x}) = f\left(x_n + (\bar{x} - x_n)\right) = f(x_n) + (\bar{x} - x_n)f'(\xi),$$

где точка ξ лежит между x_n и $\bar{x}$.

Заменив приближенно $f'(\xi)$ на $f'(x_n)$, $\bar{x}$ на x_{n+1} и выполнив преобразования, получим итерационный процесс (4). Итерации сходятся с той стороны, с которой $f(x)f''(x) > 0$. Удачным начальным приближением x_0 будет то, для которого выполняется неравенство $f(x_0)f''(x_0) > 0$. Если же $f(x_0)f''(x_0) < 0$, то мы можем либо сделать одну лишнюю итерацию для того, чтобы попасть в область монотонной сходимости (рис. 9), либо вообще не сойтись к корню, когда начальное приближение x_0 не очень хорошее (рис. 10).

Метод Ньютона можно также получить из разложения $f(x)$ по формуле Тейлора в окрестности корня $\bar{x}$:

$$0 = f(\bar{x}) = f(x_n) + (\bar{x} - x_n)f'(x_n) + \frac{1}{2}(\bar{x} - x_n)^2 f''(\xi), \quad \xi \in [x_n, \bar{x}],$$

заменив $\bar{x}$ на x_{n+1} и отбросив последний член:

$$0 = f(x_n) + (x_{n+1} - x_n)f'(x_n).$$

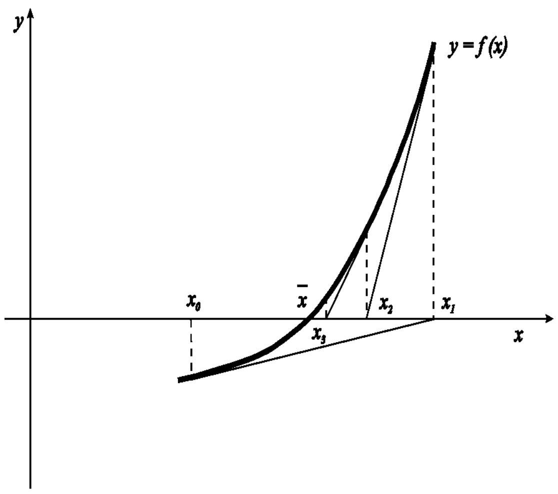


Рис. 9

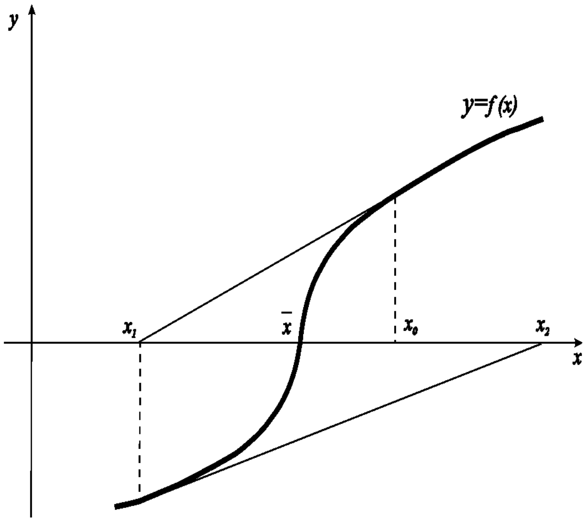


Рис. 10

Метод Ньютона можно рассматривать как частный случай метода простых итераций, если взять

$$\varphi(x) = x - \frac{f(x)}{f'(x)}.$$

Отсюда следует, что

$$\varphi'(x) = \frac{ff''}{(f')^2}.$$

Если $\bar{x}$ является простым корнем, то вблизи него $f(x) \cong a(x-\bar{x})$ и $\varphi'(x) = 0$. Для метода простой итерации достаточное условие сходимости выражается неравенством $|\varphi'(x)| < 1$. Значит, для метода Ньютона при произвольном нулевом приближении итерации сходятся, если всюду выполняется неравенство $|ff''| < (f')^2$. В противном случае сходимость будет не при любом нулевом приближении, а только в некоторой окрестности корня.

Скорость сходимости метода вблизи простого корня определяется соотношением

$$x_n - \bar{x} = \frac{1}{2}\left(x_{n-1} - \bar{x}\right)^2 \varphi''(\xi), \qquad \xi \in [x_{n-1}, \bar{x}],$$

которое получается, если в равенстве

$$x_n - \bar{x} = \varphi\left(x_{n-1}\right) - \varphi(\bar{x})$$

разложить правую часть в ряд Тейлора и учесть, что $\varphi'(\bar{x}) = 0$. Следовательно, метод Ньютона сходится с *квадратичной* скоростью (тогда как метод итерации сходится со скоростью геометрической прогрессии). Это означает, что погрешность очередного приближения вблизи простого корня примерно равна квадрату погрешности предыдущего приближения, т.е. если $(n-1)$-я итерация давала три верных знака, то n-я дает примерно шесть верных знаков, а $(n+1)$-я уже примерно 12 знаков. Геометрическая интерпретация метода Ньютона, рассматриваемого как разновидность метода простой итерации, приведена на рис. 11.

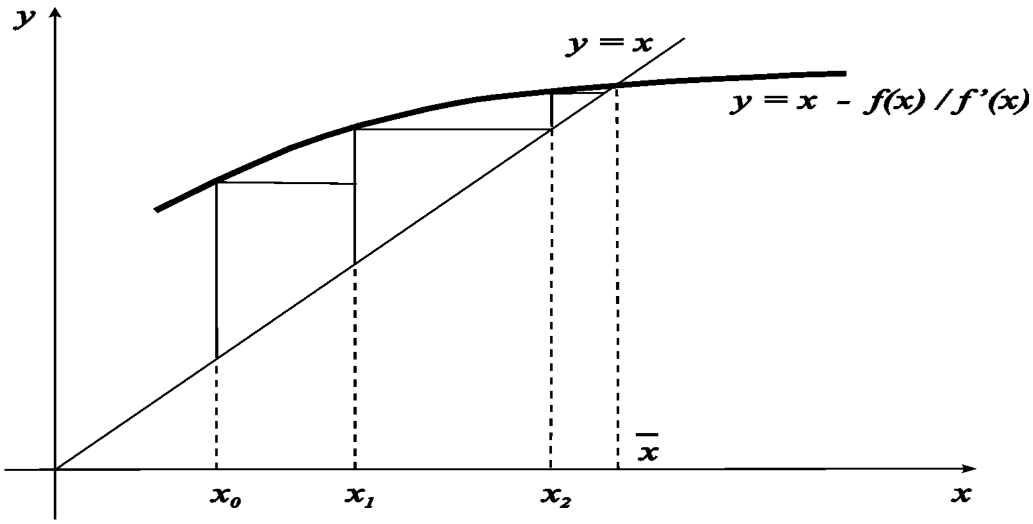


Рис. 11.

4.4. Комбинированный метод

Будем предполагать, что на отрезке $[A, B]$, $f(A)f(B) < 0$, в котором ищется корень уравнения (1), $f'(x)$ и $f''(x)$ непрерывны и сохраняют постоянные знаки.

В этих предположениях двухшаговый метод хорд (5) будет сходиться с *линейной* скоростью, если в качестве начального приближения x_0 взять точку, в которой $f(x_0)f''(x_0) > 0$,а в качестве начального приближения x_1 взять точку, в которой значение функции $f(x_1)$ имеет знак, противоположный знаку $f(x_0)$. Значениями x_0 и x_1 могут быть, например, соответствующие концы отрезка $[A, B]$. Этот метод можно рассматривать как частный случай метода простых итераций, если взять

$$\varphi(x) = x - f(x)\,\frac{(x - x_0)}{f(x) - f(x_0)}.$$

Однако в методе хорд, в отличие от метода Ньютона, производная

$$\varphi'(x) = \frac{(x_0 - \bar{x})^2}{2}\,\frac{f''(\xi)}{f(x_0)}, \quad \xi \in [x_0, \bar{x}],$$

отлична от нуля, чем и объясняется его линейная сходимость. В некоторой окрестности вблизи корня будет выполняться неравенство $|\varphi'(x)| \le q < 1$, и если точка x_1 взята из этой окрестности, то последовательность (5) будет сходиться к корню $\bar{x}$.

В случае применения комбинированного метода (6) скорость сходимости будет увеличена не только за счет использования метода Ньютона, имеющего квадратичную скорость сходимости, но и за счет того, что на каждом шаге итерационного процесса метод хорд применяется каждый раз на отрезке меньшей длины. Для комбинированного метода начальные приближения $\tilde{x}_0$ и x_0 выбираются по формулам:

— в случае $f(A)f''(A) > 0$

$$\tilde{x}_0 = A - \frac{f(A)}{f'(A)}, \qquad x_0 = \frac{Af(B) - Bf(A)}{f(B) - f(A)};$$

— в случае $f(B)f''(B) > 0$

$$\tilde{x}_0 = B - \frac{f(B)}{f'(B)}, \qquad x_0 = \frac{Af(B) - Bf(A)}{f(B) - f(A)}.$$

5. Практические критерии контроля точности

Вопрос выбора практически применимых критериев контроля точности вычисленного значения корня, т.е. определения того момента, когда следует прекращать рассмотренные итерационные процессы, требует отдельного обсуждения. Казалось бы, этот вопрос прост, поскольку сам собой напрашивается критерий, согласно которому о точности приближенно вычисленного корня $\bar{x}^*$ можно судить по тому, насколько хорошо он удовлетворяет заданному уравнению (1). При этом если число $|f(\bar{x}^*)|$ мало, то считают, что $\bar{x}^*$ является хорошим приближением для точного корня $\bar{x}$. Однако рис. 12 показывает, что это не всегда так (ε — заданная точность).

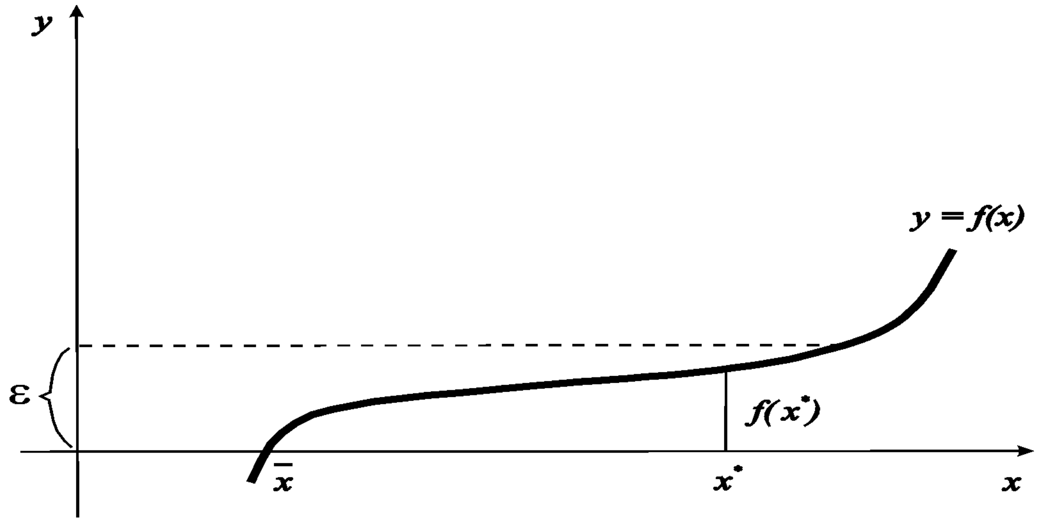


Рис. 12. $x^* = \bar{x}^*$

Если число $f|(\bar{x}^*)|$ велико, то это в свою очередь не всегда означает, что $\bar{x}^*$ является грубым приближением корня $\bar{x}$. Данное утверждение следует из рис. 13, на котором $\bar{x}^* - \bar{x} < \varepsilon$.

Иллюстрации на этих рисунках легко понять, если учесть, что при умножении уравнения (1) на произвольное число $\alpha \neq 0$ получается эквивалентное уравнение $\alpha f(x) = 0$, причем число $|\alpha f(\bar{x}^*)|$ можно сделать сколь угодно большим или сколь угодно малым за счет выбора множителя α. Таким образом, оценка значения $|f(\bar{x}^*)|$ не может служить практическим критерием приемлемости приближенно вычисленного корня $\bar{x}^*$.

Кроме этого широко распространено еще одно заблуждение: если при применении итерационных методов два последовательных приближения x_{n-1} и x_n совпадают с заданной точностью ε, то также справедливо и неравенство $|x_n - \bar{x}| < \varepsilon$. Однако в общем случае данное утверждение ошибочно. Например, для метода простой итерации возможен случай, изображенный на рис. 14.

Легко показать также, что если $\varphi'(x)$ близка к 1, то величина $|x_n - \bar{x}|$ может быть большой, тогда как значение $|x_n - x_{n-1}|$ очень мало.

Критерий оценки точности приближенного значения корня по разности между двумя его последовательными приближениями неприменим в общем случае не только для метода простой итерации, но и для любых итерационных методов, которые сходятся к корню с одной

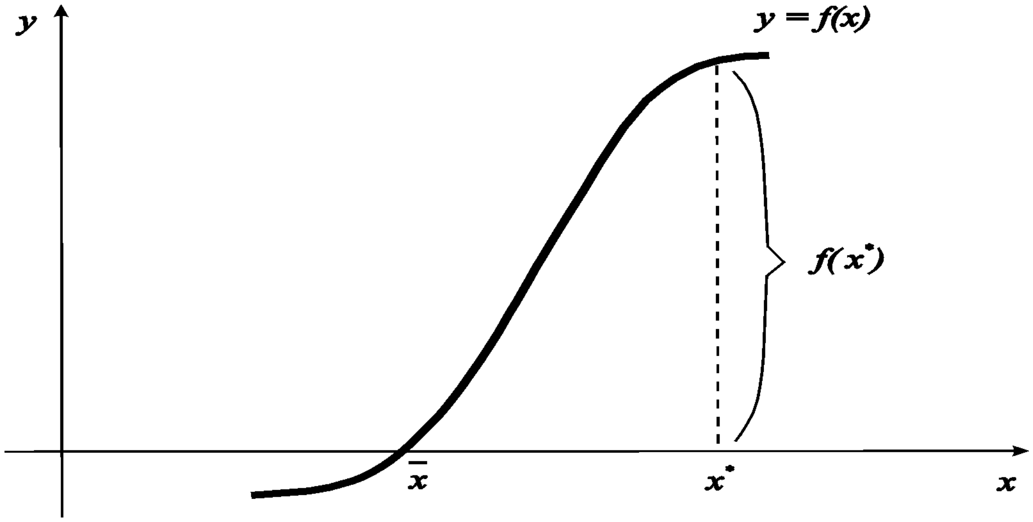


Рис. 13. $x^* = \bar{x}^*$

стороны, т.е. для односторонних итерационных методов. Например, и для метода Ньютона в общем случае совпадение с точностью до ε двух последовательных приближений x_{n-1} и x_n не гарантирует, что с той же точностью совпадут x_n и точный корень (рис. 15).

Однако этим критерием можно с успехом пользоваться на практике, когда итерации оказываются попеременно то с одной стороны, то с другой стороны корня, т.е. для двухсторонних итерационных методов. К их числу относятся метод половинного деления, комбинированный метод и метод простой итерации в случае $-1 < \varphi'(x) < 0$ (рис. 4).

Прежде чем перейти к рассмотрению практических критериев для односторонних методов, сделаем одно важное замечание, состоящее в следующем. Ни на одной из вычислительных машин невозможно представить все множество вещественных (действительных) чисел в силу ограниченности разрядной сетки машинного слова. Фактически машина оперирует с некоторым конечным множеством рациональных чисел, представленных с плавающей точкой (запятой). Поэтому необходимо учитывать различия, существующие между *идеальной* арифметикой и реальной *машинной* арифметикой. В нашем случае если математически заданная (идеальная) функция f непрерывна и меняет знак на отрезке $[A, B]$, то реально вычисляемая на машине функция принимает в действительности лишь дискретное множество значений, причем среди этих значений может не оказаться нуля функции f. Это означает, что нам следует искать не нуль функции f, а настолько малый отрезок $[\alpha, \beta]$, на котором f меняет знак, что его длина будет меньше некоторой требуемой точности ε. Если таким образом осуществляется контроль вычислений, то ε является *абсолютной* точностью. Этот способ контроля точности применим только для двухсторонних методов, поскольку для них строится последовательность вложенных отрезков, содержащих искомый нуль. При этом отрезок $[\alpha, \beta]$ можно сузить настолько, что его концевыми точками будут два соседних представимых в машине числа. Однако для односторонних методов мы не можем вести контроль по так определенной абсолютной точности, поскольку эти методы не дают последовательности вложенных сужающихся отрезков, содержащих нуль функции. В этом случае необходимо ввести некоторую *относительную* характеристику точности вычислений, например, такую: итерационный процесс проводить до тех пор, пока погрешность результата не окажется меньше длины отрезка $[A, B]$, уменьшенной в заданное количество раз. В оценках погрешности вместо длины отрезка $[A, B]$ используют расстояние между нулевым приближением и значением

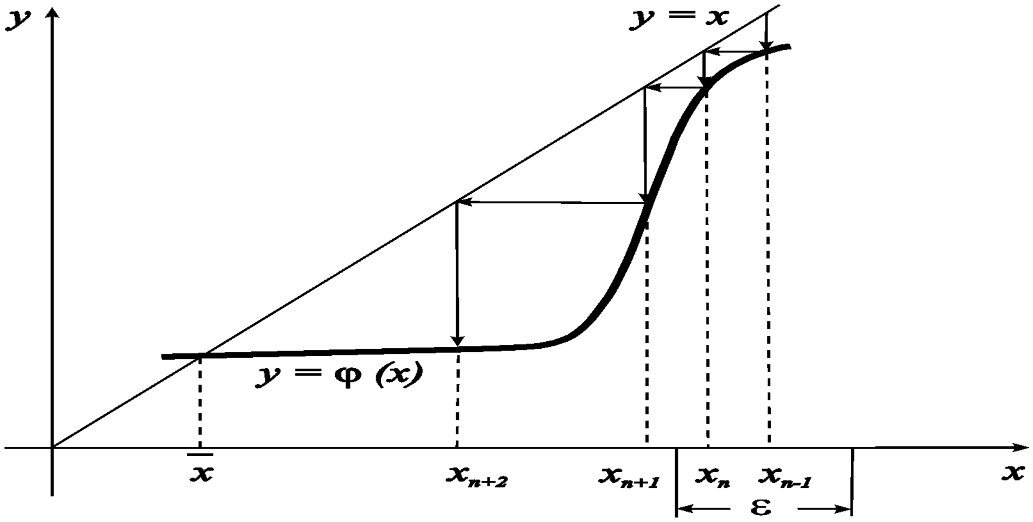


Рис. 14.

нуля, которое, правда, заранее неизвестно. В этом случае говорят, что нуль функции (корень уравнения) найден с относительной точностью ε, если вычислено такое приближение x_n, что выполнено неравенство

$$|x_n - \bar{x}| \le \varepsilon |x_0 - \bar{x}|.$$

Это означает, что итерационный процесс должен продолжаться до тех пор, пока расстояние от нулевого приближения до искомого нуля не будет уменьшено в $1/\varepsilon$ раз.

Для двусторонних методов, конечно, можно вычислять корни уравнения (1) с относительной точностью. Например, при критерии контроля точности для метода половинного деления $\beta - \alpha \le 2^{-N}(B - A)$, где N — количество итераций, говорят, что корень найден с относительной точностью 2^{-N}. В критерии $\beta - \alpha \le \varepsilon$ число ε будет абсолютной точностью. Для односторонних методов вводятся свои критерии контроля вычислений по абсолютной точности, но они отличаются от определенного выше и основаны, например, на оценках для $|x_n - \bar{x}|$.

Рассмотрим теперь практические критерии контроля точности для метода простой итерации. В случае $|\varphi'(x)| \le q < 1$ верна следующая оценка погрешности для n-го приближения к корню:

$$|x_n - \bar{x}| \le q\,|x_{n-1} - \bar{x}| \le \ldots \le q^n |x_0 - \bar{x}|.$$

Если корень $\bar{x}$ уравнения (2) ищется с относительной погрешностью $\varepsilon > 0$, чтобы при всех $n \ge n_0(\varepsilon)$ выполнялось неравенство

$$|x_n - \bar{x}| \le \varepsilon\,|x_0 - \bar{x}|, \tag{7}$$

то количество итераций, при котором выполнено данное неравенство, определяется из условия $q^n \le \varepsilon$, т.е.

$$n \ge \frac{\ln(1/\varepsilon)}{\ln(1/q)}.$$

Таким образом, минимальное количество итераций $n_0(\varepsilon)$, при котором достигается заданная относительная точность ε, равно

$$n_0(\varepsilon) = \left[\frac{\ln(1/\varepsilon)}{\ln(1/q)}\right], \tag{8}$$

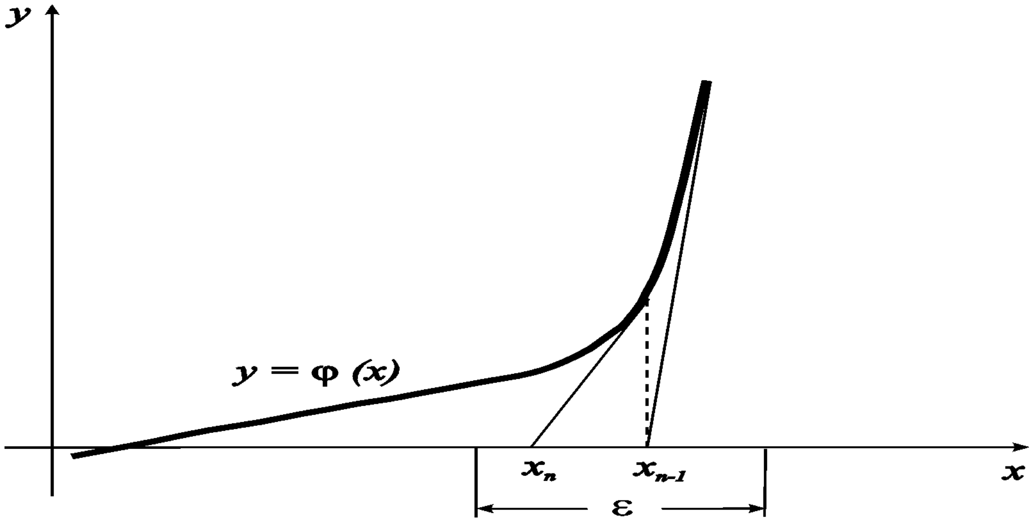


Рис. 15.

где $[a]$ — целая часть числа a. Это соотношение может быть использовано на практике для окончания итераций. Данная оценка обладает тем недостатком, что для ее применения желательно поточнее знать значение q.

Другие применимые на практике оценки имеют вид

$$|x_n - \bar{x}| \le \frac{q^n}{1-q}|x_1 - x_0|, \qquad |x_n - \bar{x}| \le \frac{q}{1-q}|x_n - x_{n-1}|, \tag{9}$$

однако также требуют знания q.

Можно использовать следующий критерий сходимости. Вблизи корня итерации сходятся примерно как геометрическая прогрессия со знаменателем

$$q = \frac{x_n - x_{n-1}}{x_{n-1} - x_{n-2}}.$$

Чтобы сумма дальнейших ее членов не превосходила заданной точности ε, то должен выполнятся такой критерий сходимости:

$$\left|q\,\frac{x_n - x_{n-1}}{1-q}\right| = \frac{(x_n - x_{n-1})^2}{|2x_{n-1} - x_n - x_{n-2}|} < \varepsilon. \tag{10}$$

При выполнении этого условия итерации (3) можно прекращать. Однако не следует забывать, что этот критерий применим вблизи корня.

Оценки (9) и (10) применяются, когда корень ищется с абсолютной погрешностью ε.

Для метода Ньютона условие окончания итераций при контроле точности по относительной ошибке определяется неравенством

$$n \ge n_0(\varepsilon) = [\log_2 \log_2 (1/\varepsilon)]. \tag{11}$$

При выполнении данного неравенства верна оценка (7). Оценка для абсолютной погрешности n-го приближения к корню $\bar{x}$ по методу Ньютона определяется неравенством

$$|\bar{x} - x_n| \le \frac{f(x_n)}{m_1}, \tag{12}$$

где m_1 — наименьшее значение $f'(x)$ на отрезке $[A, B]$. Эта оценка вытекает из формулы Лагранжа
$$f(\bar{x}) - f(x_n) = (\bar{x} - x_n)f'(\xi),$$
где ξ — промежуточное значение между $\bar{x}$ и x_n, а $f(\bar{x}) = 0$.

Метод половинного деления следует применять до тех пор, пока не выполняется неравенство $|x_n - x_{n-1}| \leq 2\varepsilon$, где ε — заданная абсолютная погрешность. Тогда середина последнего отрезка даст значения корня с требуемой абсолютной точностью.

Для комбинированного метода (6) итерационный процесс прекращается, если $|x_n - \tilde{x}_n|$ оказывается меньше ε, где ε — заданная абсолютная погрешность. В качестве значения корня целесообразно взять
$$\bar{x}^* = \frac{x_n + \tilde{x}_n}{2}.$$

6. Краткое сравнение методов

1) *Метод половинного деления.* Этот метод прост и очень надежен, поскольку обеспечивается сходимость к простому корню для любых непрерывных функций $f(x)$, в том числе недифференцируемых, при этом метод устойчив к ошибкам округления. Однако скорость сходимости, совпадающая со скоростью сходимости геометрической прогрессии со знаменателем $1/2$, невелика: за одну итерацию точность увеличивается примерно вдвое, т.е. уточнение трех цифр требует 10 итераций. Зато точность ответа гарантируется, поскольку искомый корень всегда заключен в самом внутреннем отрезке построенной последовательности, длина которого может быть сделана меньшей заданной точности. Поэтому практическая оценка погрешности вычисленного приближенного значения корня не представляет затруднений.

Перечислим недостатки метода половинного деления. Для его применения необходимо предварительно найти отрезок, на котором функция меняет знак. Если на этом отрезке лежит несколько корней, то заранее неизвестно, к какому из них сойдется процесс (хотя к одному из них заведомо сойдется). Метод неприменим для вычисления корней четной кратности. Для корней нечетной высокой кратности он сходится, но менее устойчив к ошибкам округления, возникающих при вычислении $f(x)$, и менее точен. Его нельзя, к сожалению, обобщать на случай систем уравнений.

Метод половинного деления применяется для грубого нахождения корней, так как при увеличении точности значительно возрастает объем вычислительной работы. Если каждое вычисление $f(x)$ несложно, то данный метод вполне рекомендуется для практических применений.

2) *Метод простых итераций.* Этот метод имеет важное достоинство, поскольку в нем не накапливаются ошибки вычислений (такие методы называются *самоисправляющимися*). Отдельная ошибка в вычислениях, не выводящая за пределы отрезка $[A, B]$, не повлияет на конечный результат, так как ошибочное значение можно рассматривать как новое начальное значение x_0; возможно, возрастет лишь количество итераций. Впрочем, этим свойством обладают все остальные рассмотренные методы.

В отличие от метода Ньютона и комбинированного метода в этом методе не требуется вычислять производные $f'(x)$, а соответствующим подбором $\varphi(x)$ можно добиться более быстрой сходимости по сравнению с методом половинного деления.

Укажем один довольно общий прием приведения уравнения (1) к виду (2), для которого обеспечено выполнение неравенства $|\varphi'(x)| \leq q < 1$. Пусть искомый корень $\bar{x}$ лежит на отрезке $[A, B]$, причем
$$0 < m_1 \leq f'(x) \leq M_1. \tag{13}$$

Если $f'(x)$ отрицательная, то вместо уравнения $f(x) = 0$ рассматриваем уравнение $-f(x) = 0$. Заменим уравнение (1) эквивалентным ему уравнением

$$x = \varphi(x) = x - \lambda f(x), \quad \lambda > 0.$$

Подберем параметр λ так, чтобы на $[A, B]$ выполнялись неравенства

$$0 \le \varphi'(x) = 1 - \lambda f'(x) \le q < 1. \tag{14}$$

Отсюда в силу (13) получаем

$$0 \le 1 - \lambda M_1 \le 1 - \lambda m_1 \le q.$$

Следовательно, можно выбирать $\lambda = \dfrac{1}{M_1}$, $q = 1 - \dfrac{m_1}{M_1} < 1$. Тогда неравенства (14) выполнены.

3) *Метод Ньютона.* Этот метод при $\varepsilon \to 0$ асимптотически требует меньшего количества итераций. Сходимость метода квадратичная, хотя при вычислении кратных корней он имеет скорость только геометрической прогрессии. Это обусловлено тем, что если $\bar{x}$ является p-кратным корнем, то вблизи него $f(x) \approx a(x - \bar{x})^p$ и $\varphi'(x) = \dfrac{p-1}{p}$, откуда $0 \le \varphi'(x) < 1$, поскольку $p > 1$. Суммарный объем вычислительной работы метода Ньютона порядка

$$(\alpha_1 + \alpha_2)\, \log_2 \log_2 (1/\varepsilon),$$

где α_1 и α_2 — число операций, требуемых для вычисления $f(x)$ и $f'(x)$. Однако объем вычислений будет меньше, если операции для вычисления $f(x)$ и $f'(x)$ можно совместить.

Метод Ньютона хорош еще тем, что при его помощи можно вычислять комплексные корни, если задать комплексное начальное приближение. Это же относится к методу простой итерации.

Из записи (4) следует, что чем больше численное значение $f'(x)$ в окрестности корня, тем меньше поправка к $(n+1)$-у приближению. Поэтому метод Ньютона особенно удобно применять, когда вблизи корня график функции $y = f(x)$ имеет большую крутизну. Однако если значение $f'(x)$ мало, то поправки будут велики, а значит вычисление корня может оказаться долгим или вовсе невозможным. Отсюда следует, что если кривая $y = f(x)$ вблизи точки пересечения с осью абсцисс почти горизонтальна, то применение метода Ньютона не рекомендуется.

4) *Комбинированный метод.* Недостаток односторонних методов, к которым относится методы простой итерации (в общем случае) и Ньютона, состоит в трудности контроля точности вычислений. В методе половинного деления этот недостаток отсутствует, поскольку последовательные приближения лежат по разные стороны от корня, и поэтому контроль точности можно вести по их разности. Такое преимущество двусторонних методов присуще комбинированному методу. В нем сочетается метод Ньютона и метод хорд, в которых монотонная (односторонняя) сходимость происходит по разные стороны корня, достигая при этом более быстрой сходимости, чем в методе половинного деления.

7. Этапы выполнения задания и содержание отчета

Каждый вариант задания определяется *числом*, значение которого следует вычислить в результате решения соответствующих нелинейных уравнений двумя из рассмотренных выше методов и с заданной точностью. Выполнение задания разбиваются на следующие этапы.

1) Составление уравнений для получения искомого числа. Например, пусть требуется вычислить значение $\sqrt{\pi}$ методами деления пополам и Ньютона. Тогда для вычисления приближенного значения $\bar{x}^* = \pi$ можно взять уравнение $\sin x = 0$, которое решается первым методом,

а для вычисления приближенного значения $\sqrt{\pi}$ можно взять уравнение $x^2 = \bar{x}^*$, которое затем решается методом Ньютона.

2) Определение отрезков, содержащих значения вычисляемых величин, т.е. выполнение процедур отделения корней одним из рассмотренных способов.

3) Решение составленных уравнений с заданной точностью на соответствующих отрезках; в результате должно быть получено приближенное значение искомого числа.

4) Вычисление искомого числа с помощью стандартных функций используемого языка программирования и сравнение результата с его приближенным значением (этим определяется фактическая погрешность в приближенном значении числа).

Например, число $\sqrt{\pi}$ на языке ФОРТРАН может быть получено путем вычисления следующего выражения:

$$\mathrm{SQRT}\,(4.0 * \mathrm{ATAN}(1.0))$$

5) Решение уравнений в тех же режимах по одной из программ Библиотеки численного анализа НИВЦ МГУ (http://www.srcc.msu.su/num_anal).

6) Анализ полученных результатов, в том числе сравнение составленных подпрограмм, реализующих заданные методы, с выбранной библиотечной подпрограммой по скорости и достигнутой точности.

По окончании выполнения задания составляется *отчет*, в который помимо описания результатов, полученных на перечисленных этапах, необходимо включить краткое описание заданных методов, условий и скорости их сходимости, критериев контроля точности и выбор начальных приближений, а также результаты расчетов.

8. Структура программы и результаты счета

Для выполнения задания требуется составить программу и провести по ней необходимые расчеты. Программа должна состоять из головной программы, подпрограмм, реализующих заданные методы, и подпрограмм–функций для вычисления $f(x)$, $f'(x)$ или $\varphi(x)$ в зависимости от метода. Такая модульная структура программы облегчает исследование методов для различных режимов счета и позволяет легко менять конкретный вид функций.

В головной программе задаются определенная вариантом точность вычислений, концы отрезков, содержащих значения вычисляемых величин, другие параметры, необходимые для работы подпрограмм, реализующих требуемые методы, а также осуществляются обращения к ним. Каждая такая подпрограмма имеет свою специфику, продиктованную спецификой метода, которая определяет список и смысл формальных параметров. Рассмотрим минимальный набор параметров для каждого метода, отметив, что каждая программа должна содержать в качестве формальных параметров концевые точки отрезка $[A, B]$, на котором уточняется корень, и вычисленное приближенное значение корня.

1) *Метод половинного деления.* Для этого метода реализующая его подпрограмма в качестве формальных параметров должна содержать имя подпрограммы–функции для вычисления $f(x)$, точность ε, с которой ищется корень на $[A, B]$, и оценку максимального количества итераций.

2) *Комбинированный метод.* Для этого метода реализующая его подпрограмма должна содержать те же формальные параметры, что и для метода половинного деления, только наряду с именем подпрограммы–функции для вычисления $f(x)$ следует указывать еще имя подпрограммы, вычисляющей значения как $f(x)$, так и $f'(x)$, которая, естественно, должна иметь три формальных параметра для задания значения точки x и вычисленных значений f и f' в этой точке x. Это следует сделать потому, что $f(x)$ и $f'(x)$ могут иметь общие части, которые лучше объединить в одной подпрограмме для уменьшения объема вычислительной работы при применении метода Ньютона.

Примечание. Для этих двух методов максимальное количество итераций следует задавать потому, что либо сходимости метода может не быть на данном отрезке, либо точность слишком велика и не может быть достигнута, либо заранее определен объем вычислительной работы и его нельзя превзойти. Однако может случиться так, что заданная точность оказывается достигнутой за меньшее количество итераций, поэтому в подпрограммах для этих методов следует указывать еще один формальный параметр, значение которого следует полагать равным фактически выполненному количеству итераций.

3) *Метод простой итерации.* Для этого метода реализующая его подпрограмма должна в качестве формальных параметров содержать имя подпрограммы–функции для вычисления $\varphi(x)$, относительную точность ε, с которой ищется корень на $[A, B]$, и минимальное количество итераций, определяемое по формуле (8), при котором достигается заданная относительная точность ε. Кроме того, в этой подпрограмме или в головной программе следует вычислять оценки абсолютной погрешности вычисленного корня по формулам (9) и (10), введя соответствующие формальные параметры в подпрограмму метода в зависимости от выбранного способа их вычисления. Если вариантом задания определена абсолютная точность, то следует пользоваться формулами (9) и (10).

4) *Метод Ньютона.* Для этого метода реализующая его подпрограмма должна в качестве формальных параметров содержать имя подпрограммы, вычисляющей значения как $f(x)$, так и $f'(x)$, относительную точность ε, с которой ищется корень на $[A, B]$, и минимальное количество итераций, определяемое по формуле (11), при котором достигается заданная относительная точность ε. Если вариантом задания определена абсолютная точность, то оценку погрешности вычисленного корня следует выполнять по формуле (12). Это, разумеется, можно сделать при повторных просчетах, когда значения концов отрезков (отрезка) $[\alpha_i, \alpha_{i+1}]$ известны заранее.

5) *Выбор начальных приближений.* Эта проблема для метода половинного деления отсутствует. Для остальных методов выбор начального приближения необходимо делать на основании информации, изложенной в п. 5. В список формальных параметров для подпрограмм, реализующих метод Ньютона, комбинированный метод и, возможно, метод простой итерации, следует включить также начальное приближение.

6) *Результаты счета.* Помимо вывода на экран дисплея рассмотренных выше величин, необходимо получить и вывести временны́е характеристики подпрограммы, реализующей метод. Для этого непосредственно перед оператором обращения к подпрограмме и после него следует вставить операторы обращения к специальной системной программе, которая выводит на экран время, прошедшее с момента запуска всей программы на счет до обращения к ней.

Кроме того, следует выполнить вычисление корней по одной из *стандартных библиотечных программ* при тех же условиях, что и по подпрограмме, реализующей заданный метод. Необходимо получить также временны́е характеристики работы выбранной библиотечной подпрограммы.

Необходимо также предусмотреть вывод на экран разности между двумя последними последовательными приближениями к корню и значения функции для приближения, выданного в качестве корня.

Разумеется, все действия по выводу результатов счета и использованию библиотечных подпрограмм следует выполнять в головной программе.

7) *Контроль ошибок при счете.* При использовании подпрограммы, реализующей итерационный метод, можно ошибиться в задании входных параметров или же оказаться вне области сходимости метода. Может так случиться, что сходимость к корню будет медленной и выделенный на его нахождение объем вычислительной работы в какой-то момент счета по

программе оказывается исчерпанным. Поэтому подпрограмму следует составить как можно более *робастной*, т.е. способной выявлять и, по возможности, исправлять все выявленные ошибочные ситуации, выводящие из области ее применимости.

Для этого в подпрограмме следует предусмотреть проверку условия $f(A)f(B) < 0$, а для двухсторонних методов дополнительно проверку, не сделано ли больше итераций, чем их заданное максимальное количество. В случае выявления ошибочной ситуации следует выводить соответствующие диагностические сообщения и вернуться в головную программу. Операторы вывода диагностических сообщений не следует непосредственно помещать в подпрограмму; необходимо их выделить в отдельную подпрограмму диагностических сообщений.

9. Варианты задания

1) Задачи на нахождение чисел.

1. π^e	7. $\arcsin\left(\sqrt{\pi} - \dfrac{e}{2}\right)$
2. $-\sqrt[4]{\pi} + 1$	8. $\arccos\ \sqrt[-1]{e}$
3. $\sqrt[4]{e} + 1$	9. $4\left(\operatorname{arctg}\dfrac{1}{5} + \operatorname{arctg}\dfrac{1}{239}\right)$
4. e^{π}	10. $\arcsin\dfrac{\pi}{17}$
5. $\arcsin\left(\dfrac{e}{\pi}\right)$	11. $\arccos\left(\dfrac{1}{8}\arcsin\sqrt{\dfrac{e}{\pi}}\right)$
6. $\arccos\left(\dfrac{-\pi}{e^2}\right)$	12. $\sin\pi$

2) Методы:

а) деление пополам;

б) простая итерация;

в) Ньютона;

г) комбинированный.

3) Точность:

а) абсолютная;

б) относительная;

в) максимально достижимая на используемой ЭВМ (см. п. 12).

10. Указания по выполнению задания на ЭВМ

В качестве стандартной библиотечной подпрограммы следует взять одну из подпрограмм ZF10R, ZF11R или ZF15R на языках Фортран или Си, входящих в состав Библиотеки НИВЦ МГУ для решения типовых задач численного анализа (http://www.srcc.msu.su/num_anal).

Список параметров подпрограммы ZF15R следующий:

$$\mathrm{ZF15R(F, A, B, EPS, ROOT, ITMAX, IERR)}$$

Параметры

F — имя вещественной подпрограммы-функции вычисления $f(x)$ в любой точке интервала (A, B);

A, B — заданная нижняя и верхняя границы интервала, на котором $f(x)$ меняет знак (тип: вещественный);

EPS — заданная абсолютная погрешность вычисления нуля функции (тип: вещественный);

ITMAX — целая переменная, значение которой перед началом работы подпрограммы должно быть положено равным максимальному числу итераций, ориентировочно требуемых для обеспечения сходимости; ее значение в результате работы подпрограммы полагается равным действительному числу итераций, потребовавшихся для обеспечения сходимости;

ROOT — вещественная переменная, значение которой в результате работы подпрограммы полагается равным вычисленному нулю функции $f(x)$;

IERR — целая переменная, служащая для сообщения об ошибках, обнаруженных в ходе работы подпрограммы; при этом

IERR = 65 — когда нуль функции не может быть подсчитан в пределах заданного числа итераций;

IERR = 66 — когда функция не меняет знака на заданном интервале.

Подпрограмма ZF15R применяет метод половинного деления (метод бисекций) в комбинации с *методом секущих*. Бисекция осуществляется в тех случаях, когда использование метода секущих не приводит к уменьшению в два раза наименьшего вычислительного значения функции или выводит за границы текущего суженного отрезка. Если в заданных при обращении к подпрограмме границах отрезка $[A, B]$ функция имеет одинаковые знаки, т.е. $f(A)f(B) > 0$, то делается попытка методом бисекций за заданное число итераций найти такой отрезок $[\alpha, \beta]$ с $[A, B]$, на котором $f(A)f(B) < 0$. Нуль функции считается вычисленным с заданной абсолютной точностью ε, если найдена такая точка $x \in [A, B]$, что $|f(x)| < \varepsilon$. Данную подпрограмму практически удобно применять лишь для грубого нахождения нулей заданной функции, поскольку при увеличении точности может значительно вырасти объем вычислительной работы. Поэтому в ZF15R применяется только этот критерий сходимости, удовлетворение которому, вообще говоря, не гарантирует достижения заданной точности.

Список параметров подпрограммы ZF11R следующий:

$$\mathrm{ZF11R(F, A, B, EPS, NDIG, ITMAX, ROOT, IERR)}$$

Параметры

F — имя вещественной подпрограммы-функции вычисления $f(x)$ в любой точке интервала (A, B);

A, B — заданная нижняя и верхняя границы интервала, на котором $f(x)$ меняет знак (тип: вещественный);

EPS — первый критерий сходимости: заданная абсолютная погрешность вычисления нуля функции (тип: вещественный);

NDIG — второй критерий сходимости: заданное число значащих цифр, с которыми предполагается вычислить нуль функции (тип: целый);

ITMAX — целая переменная, значение которой перед началом работы подпрограммы должно быть положено равным максимальному числу итераций, ориентировочно требуемых для обеспечения сходимости; в результате работы подпрограммы ее значение полагается равным действительному числу итераций, потребовавшихся для обеспечения сходимости в соответствии с заданными критериями;

ROOT — вещественная переменная, значение которой в результате работы подпрограммы полагается равным вычисленному нулю функции;

IERR — целая переменная, служащая для сообщения об ошибках, обнаруженных в ходе работы подпрограммы; при этом

IERR = 65 — когда нуль функции не может быть подсчитан в пределах заданного числа итераций;

IERR = 66 — когда функция не меняет знака на заданном интервале.

В подпрограмме ZF11R реализован метод секущих.

Список параметров подпрограммы ZF10R следующий:

$$\mathrm{CALL\ ZF10R(F, A, B, EPS, NDIG, ITMAX, ROOT, IERR)}$$

Параметры

F — имя вещественной подпрограммы-функции вычисления $f(x)$ в любой точке интервала (A, B);

A, B — заданная нижняя и верхняя границы интервала, на котором $f(x)$ меняет знак (тип: вещественный);

EPS — первый критерий сходимости: заданная абсолютная погрешность вычисления нуля функции (тип: вещественный);

NDIG — второй критерий сходимости: заданное число значащих цифр, с которыми предполагается вычислить нуль функции (тип: целый);

ITMAX — целая переменная, значение которой перед началом работы подпрограммы должно быть положено равным максимальному числу итераций, ориентировочно требуемых для обеспечения сходимости; ее значение в результате работы подпрограммы полагается равным действительному числу итераций, потребовавшихся для обеспечения сходимости в соответствии с заданными критериями;

ROOT — вещественная переменная, значение которой в результате работы подпрограммы полагается равным вычисленному нулю функции $f(x)$;

IERR — целая переменная, служащая для сообщения об ошибках, обнаруженных в ходе работы подпрограммы; при этом

IERR = 65 — когда нуль функции не может быть подсчитан в пределах заданного числа итераций;

IERR = 66 — когда функция не меняет знака на заданном интервале.

Подпрограмма ZF10R использует двусторонний метод Брента (см. п. 16), основанный на комбинации методов половинного деления, секущих и *обратной квадратичной интерполяции* (см. п. 15). Этот метод не так чувствителен к начальным приближениям и в предположении, что в окрестности искомого нуля функция $f(x)$ является непрерывно дифференцируемой, имеет более чем линейную сходимость.

В подпрограммах ZF11R и ZF10R применяются два критерия сходимости. Пусть x_{n-1} и x_n являются двумя последовательными приближениями к нулю $\bar{x}$ функции $f(x)$. Тогда точка x_n принимается за искомый нуль, если выполнен один из двух критериев сходимости: $|f(x_n)| \le \varepsilon$ или

$$\frac{x_n - x_{n-1}}{\max\left(|x_n|, |x_{n-1}|, 0.1\right)} \le 10^{-l},$$

где ε — заданная абсолютная погрешность, а l — заданное число значащих цифр в приближенном значении нуля. При обращении к этим подпрограммам может быть задан либо только один из этих критериев, либо оба критерия одновременно. Хотя такой контроль сходимости в общем случае не гарантирует для подпрограммы ZF11R достижения требуемой точности (см. п. 6), однако для реальных задач он вполне применим. Задание второго критерия в ZF10R теоретически обосновано, поскольку реализованный в ней метод Брента является двусторонним и два последовательных приближения x_{n-1} и x_n лежат по разные стороны от искомого нуля.

ДОПОЛНЕНИЕ

11. Еще раз о практических критериях контроля точности и окончания счета

Как указывалось выше, окончание счета для двухсторонних методов осуществляется тогда, когда либо заданная абсолютная точность оказывается достигнутой, либо выполнено заданное разрешенное максимальное количество итераций, а для односторонних методов— когда выполнено минимальное количество итераций, при котором достигается требуемая относительная точность. К этим критериям окончания счета можно было бы попытаться добавить какой-либо другой критерий (например, $\dfrac{|x_{n+1}-x_n|}{|x_n-x_{n-1}|}>1$), который позволял бы выявлять расходимость метода на ранних этапах итерационного процесса и тем самым давал бы заметную экономию вычислительной работы. Однако применение такого рода критериев практически не оправдано из-за того, что они часто приводят к преждевременному завершению итерационного процесса, поскольку итерации могут вначале расходиться, прежде чем начать сходиться к корню.

Рассмотрим теперь еще один критерий контроля точности для двухсторонних методов, который можно назвать *комбинированным*, поскольку он объединяет контроль по абсолютной ε_A и относительной ε_O погрешностям:

$$|x_{n+1}-x_n| \le \varepsilon_A + \varepsilon_O\,|x_n|. \tag{15}$$

В пользу этого критерия можно высказать следующие соображения.

Если задана только допустимая абсолютная погрешность ε_A (т.е. $\varepsilon_O = 0$), то тем самым фиксируется разряд приближенного значения корня, соответствующий требуемой самой младшей верной цифре этого значения. Однако если задать абсолютную погрешность без учета величины порядка искомого корня и длины разрядной сетки машины, то контроль вычислений по абсолютной погрешности может оказаться невозможным. Например, если вычисления проводятся с семью десятичными разрядами и искомый корень равен 55555.55, то задание абсолютной погрешности, равной 10^{-4} является бессмысленным и приведет к зацикливанию итерационного процесса. Поэтому если мы хотим,чтобы четвертый разряд приближенного значения корня соответствовал самой младшей верной цифре, то в данном примере мы должны положить абсолютную погрешность равной 10, что в отрыве от величины порядка искомого корня и количества разрядов, с которыми проводятся вычисления, может показаться нелепым, поскольку обычная абсолютная погрешность используется для задания количества *верных цифр после точки* (запятой), отделяющей целую часть от дробной. Таким образом, чтобы разумно задать абсолютную погрешность, нужно предварительно знать величину порядка искомого решения задачи (а это, конечно, не всегда возможно) и учитывать величину начального приближения.

Если задана только относительная погрешность ε_O (т.е. в (15) $\varepsilon_A = 0$), то тем самым фиксируется общее требуемое количество верных цифр приближенного значения корня. Однако если искомый корень мал и значение x_n становится слишком близким к нулю, то даже при разумном задании ε_O неравенство (15) может никогда не достигаться или же при вычислении $\varepsilon_O\,|x_n|$ может произойти образование машинного нуля (потеря значимости). Поясним, почему неравенство (15) может никогда не достигаться, даже если в машинном представлении произведение $\varepsilon_O\,|x_n|$ не равно нулю, и итерационный процесс гарантировано сходится (как, например, в методе половинного деления). Это следует из следующего *фундаментального свойства* систем представления на машинах чисел с плавающей точкой (запятой): расстояние между числом x и соседним по отношению к нему числом не меньше $\dfrac{\mathrm{macheps}\cdot|x|}{\beta}$ и не больше macheps $\cdot |x|$, если только само число или соседнее число не равно нулю. Здесь β — основание системы счисления машины, а машинно-зависимый параметр macheps, называемый *машинным эпсилон*, характеризует относительную точность плавающей арифметики и является наименьшим числом с плавающей точкой, для которого удовлетворяется неравенство

$$1.0 + \text{macheps} > 1.0.$$

Значение macheps$/\beta$ равно расстоянию от 1.0 до левого соседнего числа, а само значение macheps $= \beta^{1-t}$ (t — количество разрядов в *мантиссе* машинного слова) представляет собой расстояние от 1.0 до правого соседнего числа. Таким образом, если $\varepsilon_O\,|x_n|$ окажется меньше macheps $\cdot |x_n|/\beta$, то неравенство (15) при $\varepsilon_A = 0$ никогда не будет достигаться, а основанный на нем итерационный процесс никогда не завершится. Если мы хотим, чтобы x_{n+1} и x_n стали максимально близкими друг другу, т.е. соседними числами, то критерий контроля точности должен быть таким:

$$|x_{n+1} - x_n| \le \text{macheps} \cdot \max\left(|x_{n+1}|, |x_n|\right).$$

Данный критерий неприменим для небольшой окрестности нуля, в которой происходит образование машинного нуля при вычислении правой части. Отметим, что расстояние от нуля до правого (левого) соседнего числа не связано с параметром macheps и представляет собой самостоятельный машинно-зависимый параметр.

Таким образом, применение неравенства (15) позволяет избегать тех тупиковых ситуаций, которые могут возникнуть, если задавать требуемое количество верных знаков в приближенном решении, не заботясь о величине его порядка.

Как указывалось выше, применение критерия

$$|f(x_n)| \le \tilde{\varepsilon}_A \qquad (16)$$

не гарантирует, что приближение x_n к искомому корню найдено с предписанной точностью; то же самое можно сказать и о критерии (15) для односторонних методов. Однако на практике для односторонних методов одновременное применение критериев (15) и (16) вполне разумно, если значения ε_A, ε_O и $\tilde{\varepsilon}_A$ выбрать достаточно малыми.

12. Как лучше вычислять среднее арифметическое двух чисел

Ошибки округления, которые сами по себе кажутся незначительными, могут оказать существенное влияние на конечный результат, если для его получения выполняется большое количество арифметических операций. Поэтому следует стараться минимизировать ошибки в каждой операции или в последовательности операций, уменьшая тем самым их распространение и воздействие на конечный результат. Рассмотрим как это можно сделать в случае выполнения простой операции вычисления среднего арифметического $\dfrac{a+b}{2}$ двух чисел a и b, т.е. операции вычисления очередного приближения метода половинного деления.

Возможны два способа выполнения этой операции:

$$c = \frac{a+b}{2} \qquad (17) \qquad \text{и} \qquad c = a + \frac{b-a}{2}. \qquad (18)$$

Очевидно, что формула (17) требует на одну операцию сложения меньше, чем формула (18), но с точки зрения точности не всегда лучше. Действительно, пусть вычисления выполняются в десятичной арифметике с тремя цифрами и с округлением для $a = 0.596$ и $b = 0.600$. Тогда будем иметь

$$c = \frac{0.596 + 0.600}{2} = \frac{1.200}{2} = 0.600,$$

хотя правильное значение c равно 0.598. Если же мы будем проводить вычисления по формуле (18), то будем иметь

$$c = 0.596 + \frac{0.600 - 0.596}{2} = 0.596 + \frac{0.004}{2} = 0.598.$$

Отметим, что в данном примере, для которого формула (18) является предпочтительней, числа a и b имеют одинаковые знаки.

Рассмотрим другой пример для десятичной четырехзначной арифметики, в которой вместо операции округления применяется отбрасывание лишних разрядов. Пусть $a = -3.483$ и $b = 8.765$. Тогда по формуле (17) будем иметь

$$c = \frac{-3.483 + 8.765}{2} = \frac{5.282}{2} = 2.641,$$

что представляет собой точный результат. Вычисления по формуле (18) дают

$$c = -3.483 + \frac{8.765 + 3.483}{2} = -3.483 + \frac{12.24}{2} = -3.483 + 6.120 = 2.637.$$

Даже если бы мы здесь выполняли округление, то все равно результат по формуле (18) отличался бы от точного, поскольку значение c равнялось бы 2.642. Следовательно, в данном примере, в котором числа a и b имеют разные знаки, формула (17) оказалась предпочтительнее.

Отсюда мы можем сделать вывод, что для достижения наивысшей точности мы должны применять либо формулу (17), либо формулу (18) в зависимости от знаков a и b. Поэтому наилучшую формулу для вычисления среднего арифметического a и b можно представить следующим образом:

$$\text{если} \quad (\operatorname{sign}(a) \neq \operatorname{sign}(b)), \quad \text{то} \quad c = \frac{a+b}{2} \quad \text{иначе} \quad c = a + \frac{b-a}{2}.$$

Рассмотренный пример показывает, какие меры предосторожности следует применять при реализации простейших операций и как выбор правильных формул может улучшить точность программы.

Вернемся к методу половинного деления. Применение формулы (18) для него предпочтительней, поскольку в конечном счете наступает такой момент, когда знаки приближения a_k и b_k к корню будут одинаковыми вплоть до окончания итерационного процесса (исключением является случай, когда искомый корень равен нулю). До этого момента, когда точность формулы (17) выше, нет необходимости минимизировать ошибки округления, поскольку их влияние незначительно по сравнению с общей ошибкой в текущих приближениях к корню.

Рассмотренный способ уточнения вычислений средних арифметических чисел, по крайней мере для метода половинного деления, может показаться надуманным, поскольку, как правило, вовсе не нужна столь высокая точность, а данный метод является самоисправляющимся (т.е. отдельная ошибка в вычислениях не влияет на результат, если она не выводит за пределы отрезка $[A, B]$ и точность не слишком высока, потому что ошибочное значение можно считать новым приближением; возможно возрастет лишь количество итераций). Однако это не так. Формулу (18) следует применять в программах, которые могут быть потенциально использованы для вычисления корней с максимально возможной точностью, Кроме того, на машинах, в которых вместо округления осуществляется отбрасывание лишних разрядов, формула (17) может вывести за пределы текущего подотрезка (например, в десятичной трехзначной арифметике для $a = 0.982$ и $b = 0.987$ будем иметь:

$$c = \frac{a+b}{2} = \frac{0.982 + 0.987}{2} = \frac{1.96}{2} = 0.980.$$

И, наконец, лучше придерживаться общего правила, согласно которому новое приближение следует вычислять прибавлением полученной поправки к текущему приближению.

13. Как лучше проверить, меняет ли функция знак на концах отрезка

Обычный способ проверки изменения знака функции $f(x)$ на концах отрезка $[A, B]$ состоит в проверке условия

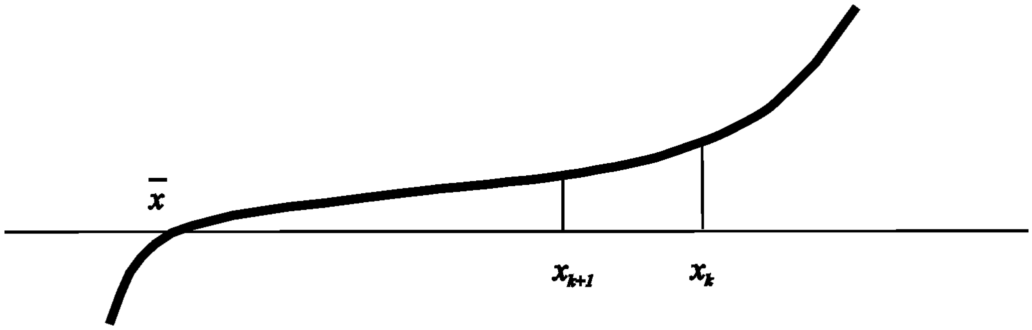
$$f(A)f(B) < 0.$$

Однако если значение $f(A)$ и $f(B)$ близки к нулю, то результат умножения $f(A)f(B)$ может стать машинным нулем, и этот способ окажется неприемлемым. Поэтому требуемую проверку лучше выполнять следующим образом:

$$f(A) * \frac{f(B)}{\mathrm{abs}(f(B))} < 0.$$

14. Неустойчивость корней

Рассмотрим случай, когда функция $f(x)$ вблизи корня $\bar{x}$ имеет небольшую производную, т.е. ее график почти горизонтален в окрестности корня.



Очевидно, что тогда любое небольшое возмущение функции $f(x)$ (скажем, $f(x) + \varepsilon$) приведет к заметному изменению в значении корня $\bar{x}$. Подобная неустойчивость характеризует *корень* $\bar{x}$, а не саму функцию $f(x)$, поскольку она может иметь другие корни, не столь чувствительные к возмущениям, и типична для полиномов. Например, рассмотрим уравнение $(x-1)^2 = 0$, и пусть его правая часть оказалась возмущенной на 10^{-6}, т.е. в действительности уравнение имеет вид $(x-1)^2 = 10^{-6}$. Тогда корни уравнения будут равны 1 ± 10^{-3}, т.е. возмущение свободного члена полинома на 10^{-6} вызвало изменение в корнях, равное 10^{-3}. Такая чувствительность корней полиномов к возмущениям в их коэффициентах еще более выражена у полиномов высоких степеней.

Практическая проверка устойчивости вычисленного корня состоит в повторном просчете для возмущенной $f(x)$ с другой точностью и анализе величины изменения нового корня по сравнению с величиной возмущения.

15. Еще раз о методе Ньютона и его модификациях

Как указывалось в п. 7, если $\bar{x}$ является p-кратным корнем, то вблизи него метод Ньютона имеет скорость геометрической прогрессии со знаменателем $\dfrac{p-1}{p}$. Поскольку производная $f'(x_n)$ мала, когда приближение x_n близко к кратному корню, то может появиться опасение, не возникнет ли переполнение арифметического процессора машины при вычислении поправки $\dfrac{f(x_n)}{f'(x_n)}$. Однако переполнения произойти не может. Действительно, пусть $\bar{x}$ — корень кратности уравнения $f(x) = 0$. Тогда мы можем записать, что $f(x) = (x - \bar{x})^3\, q(x)$, где $q(\bar{x}) \neq 0$. Если обозначить $x_n - \bar{x} = \varepsilon$, то по расчетной формуле метода Ньютона будем иметь:

$$x_{n+1} = x_n - \frac{\varepsilon^3 q(x_n)}{3\varepsilon^2 q(x_n) + \varepsilon^3 q'(x_n)} = x_n - \frac{\varepsilon}{3} + O(\varepsilon^2).$$

Таким образом, хотя значение знаменателя $f'(x_n)$ мало вблизи кратного корня, значение числителя все же меньше на множитель ε, и поэтому операция деления не вызывает затруднений.

Если производная $f'(x)$ мало меняется в окрестности корня, то получаем видоизмененный метод Ньютона

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}.$$

Геометрически этот способ означает, что каждый раз касательные проводятся параллельно касательной к кривой $y = f(x)$ в точке $(x_0, f(x_0))$ (рис. 16).

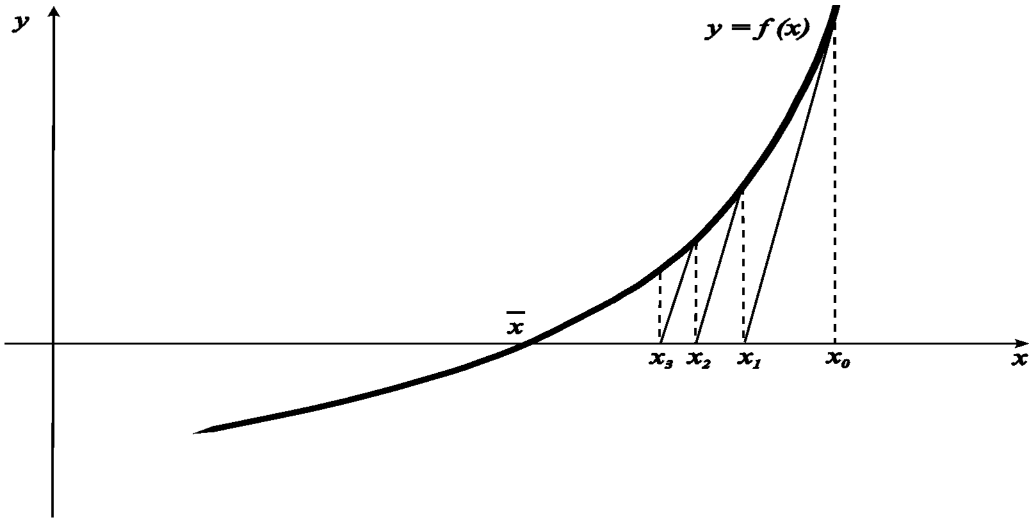


Рис. 16

Это видоизменение позволяет не вычислять $f'(x_n)$ при каждой новой итерации, что полезно, если $f'(x_n)$ трудоемка для вычислений. Сходимость процесса обеспечивается при тех же условиях, что и для метода Ньютона.

Можно вообще избежать вычисления производной, если заменить ее первой разделенной разностью, найденной по двум последним итерациям. Это означает, что касательная заменяется секущей. В этом случае вместо метода Ньютона имеем следующий итерационный процесс, получивший название метода *секущих*:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\, f(x_n).$$

В данном процессе для вычисления очередного приближения необходимо знать два предыдущих. Такие процессы называются *двухшаговыми*. Геометрическая интерпретация приведена на рис. 17.

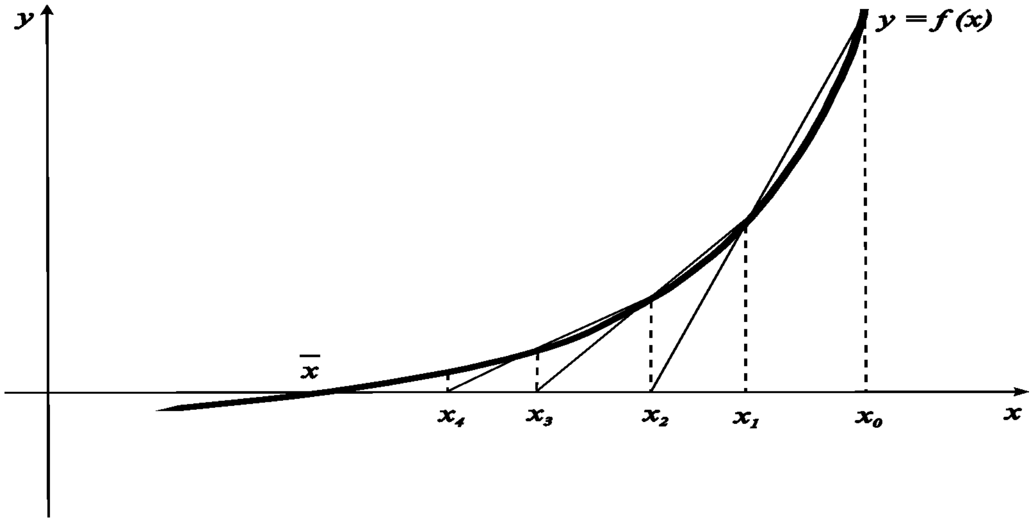


Рис. 17

Скорость сходимости метода секущих вблизи корня $\bar{x}$ определяется соотношением:

$$x_{n+1} - \bar{x} \approx (x_n - \bar{x})^{1.62} \left(\frac{f''(\bar{x})}{f'(\bar{x})} \right)^{0.62}.$$

Отсюда видно, что в методе Ньютона ошибка убывает быстрее, поскольку у него скорость сходимости квадратичная. Однако в методе Ньютона приходится считать как значение функции, так и значение производной, а в методе секущих — только значения функции. Сравнение объемов вычисления этих двух методов можно сделать следующим образом. Для метода секущих минимальное количество итераций, требуемых для достижения заданной относительной точности ε, определяется соотношением

$$n_0(\varepsilon) \sim 1.44 \log_2 \log_2 \frac{1}{\varepsilon}.$$

Обозначим через α_1 объем вычислительной работы, требуемой для вычисления значения функции в точке, а через α_2 — дополнительный объем для вычисления значения производной в той же точке (при этом учитывается, что $f(x)$ и $f'(x)$ могут иметь общие выражения, которые следует вычислять на каждой итерации только один раз). Тогда суммарные затраты работ для метода секущих будут порядка

$$1.44\, \alpha_1 \log_2 \log_2 \frac{1}{\varepsilon},$$

а для метода Ньютона

$$(\alpha_1 + \alpha_2) \log_2 \log_2 \frac{1}{\varepsilon}.$$

Если совмещение операций для вычисления $f(x)$ и $f'(x)$ таково, что $\alpha_2 < 0.44\alpha_1$, то с точки зрения затрат вычислительной работы при наличии хорошего начального приближения предпочтительнее использовать метод Ньютона.

В знаменателе метода секущих стоит разность значений функции, что приводит к потере значащих цифр вблизи корня (особенно корня высокой кратности), поскольку эти значения становятся малыми и очень близкими. Возникает "разболтка" счета, вследствие которой ограничивается точность, с которой можно найти корень. Такого рода "разболтку" учитывают применением следующего приема. Выбирают некоторое не очень малое ε и ведут итерации до тех пор, пока не выполнится неравенство $|x_{n+1} - x_n| < \varepsilon$. Затем расчеты продолжаются, пока убывает $|x_{n+1} - x_n|$. Первое же возрастание разности означает, как правило, начало "разболтки" после чего итерации прекращают. Этот способ называют *способом Гарвика.*

Следует отметить, что в библиотечных подпрограммах предпочтение все же отдается методу секущих перед методом Ньютона именно в силу того, что для метода секущих не нужно задавать операторов вычисления $f'(x)$.

16. Метод парабол

Этот метод называют еще *методом Мюллера*, или *квадратичной интерполяцией.*

Рассмотренный выше метод секущих основан на замене функции $f(x)$ интерполяционным многочленом первой степени, проведенным по узлам x_n и x_{n-1}. По трем последним итерациям можно построить интерполяционный многочлен второй степени, т.е. заменить график функции $f(x)$ *параболой* $y = ax^2 + bx + c$. Коэффициенты параболы a, b и c определяются из условия, что парабола должна проходить через точки

$$\begin{array}{lll} (x_{n-2}, f_{n-2}), & (x_{n-1}, f_{n-1}), & (x_n, f_n), \\ f_i = f(x_i), & i = n-2, n-1, n. & \end{array}$$

Поэтому имеем следующие расчетные формулы для их определения, которые получаются из решения трех линейных систем с тремя неизвестными:

$$\begin{aligned} a &= \frac{(f_n - f_{n-1})(x_{n-1} - x_{n-2}) - (f_{n-1} - f_{n-2})(x_n - x_{n-1})}{(x_n - x_{n-1})(x_{n-1} - x_{n-2})(x_n - x_{n-2})}; \\ b &= -a(x_n + x_{n-1}) + \frac{f_n - f_{n-1}}{x_n - x_{n-1}}; \\ c &= f_n - a x_n^2 - b x_n. \end{aligned}$$

Для нахождения очередного приближения x_{n+1} вычисляются корни квадратного уравнения $ax^2 + bx + c = 0$, которые имеют вид $u \pm \sqrt{v}$. В качестве x_{n+1} выбирается тот корень, у которого знак перед радикалом совпадает со знаком числа u. Такой выбор делается для того, чтобы избежать потери значимости при вычитании двух возможно близких величин u и $\sqrt{v}$.

Данный итерационный процесс является *трехшаговым* и требует для начала расчетов задания трех первых начальных приближений. Обычно библиотечные программы требуют задания только одного начального приближения к искомому корню; два других приближения генерируются самими программами, поскольку их точное расположение на числовой оси не столь существенно (например, если $x_0 = \gamma$, то в качестве x_1 и x_2 можно выбрать 0.9γ и 1.1γ соответственно).

Скорость сходимости метода парабол вблизи простого корня определяется соотношением

$$|x_{n+1} - \bar{x}| \approx \left| \frac{f'''(\bar{x})}{6f'(\bar{x})} \right|^{0.42} |x_n - \bar{x}|^{1.84},$$

т.е. его скорость сходимости даже медленнее квадратичной, хотя этот метод построен по образцу методов третьего порядка. Этот факт объясняется тем, что замена графика функции параболой означает построение интерполяционного многочлена второй степени в форме Ньютона

$$P_2(x) = f(x_n) + (x - x_n) f(x_n; x_{n-1}) + (x - x_n)(x - x_{n-1}) f(x_n; x_{n-1}; x_{n-2}),$$

где

$$f(x_n;x_{n-1}) = \frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}} \quad - \text{ разделенная разность первого порядка,}$$

$$f(x_n;x_{n-1};x_{n-2}) = \frac{f(x_n;x_{n-1})-f(x_{n-1};x_{n-2})}{x_n-x_{n-2}} \quad - \text{ разделенная разность второго порядка.}$$

Таким образом, существенное уменьшение скорости сходимости связано с заменой производных разделенными разностями. Вблизи кратного корня сходимость еще медленнее, хотя и более быстрая, чем линейная. Строить аналогичные методы с использованием интерполяционных многочленов более высокой степени невыгодно, поскольку расчеты по ним усложняются, а скорость сходимости не будет быстрей квадратичной.

Метод парабол обладает рядом важных достоинств. Прежде всего, практика его использования подтверждает, что для преобладающего большинства решаемых задач этот метод сходится к корню почти при любом наборе начальных приближений, причем для него не нужно выполнение предварительной процедуры отделения корней. Метод парабол исключительно эффективен для нахождения всех корней многочленов высоких степеней. Кроме того, корни квадратного уравнения $u \pm \sqrt{v}$ могут быть комплексными (т.е. ветви параболы могут не пересекать вещественной оси), хотя все предыдущие приближения могут быть вещественными. Это означает, что метод парабол обладает возможностью вычисления *комплексных* корней функции, даже если начальные приближения являются вещественными. Метод парабол обладает тем недостатком, что "разболтка" счета вблизи корня сказывается еще сильнее, т.к. в расчетных формулах используются вторые разделенные разности. Однако корни все же можно найти с хорошей точностью, а для определения оптимального количества итераций следует пользоваться способом Гарвика, который описан в п. 14.

Если ищутся только вещественные корни, то метод парабол неудобен тем, что его реализация должна быть выполнена в комплексной арифметике, чтобы учесть возможность появления комплексных приближений x_n. Поэтому и подпрограмма вычисления $f(x)$ должна быть написана в комплексной арифметике. Поскольку арифметические операции для комплексных чисел требуют по крайней мере в два раза больше времени, чем для вещественных чисел, то при поиске только вещественных корней очевидна потеря эффективности. Чтобы избежать этого, применяют метод *обратной квадратичной интерполяции.* Существо этого метода проще всего пояснить следующим образом.

В (прямом) методе парабол мы строили по трем точкам такую параболу, ветви которой направлены либо вниз, либо вверх, причем, очевидно, ветви могут и не пересекаться с вещественной осью, что и определяет возможность вычисления комплексных корней. Геометрически этот метод иллюстрируется на рис. 18.

Однако по тем же трем точкам (x_{n-2}, f_{n-2}), (x_{n-1}, f_{n-1}), (x_n, f_n) мы можем построить параболу

$$x = ay^2 + by + c,$$

ветви которой направлены либо вправо, либо влево вдоль вещественной оси, а это означает, что одна ветвь *обязательно* пересечется с вещественной осью, и очередное приближение к корню *всегда* будет вещественным (рис. 19).

Метод, основанный на построении такого рода парабол, называется *методом обратной интерполяции.* Очевидно, что в качестве очередного приближения в этом методе выбирается

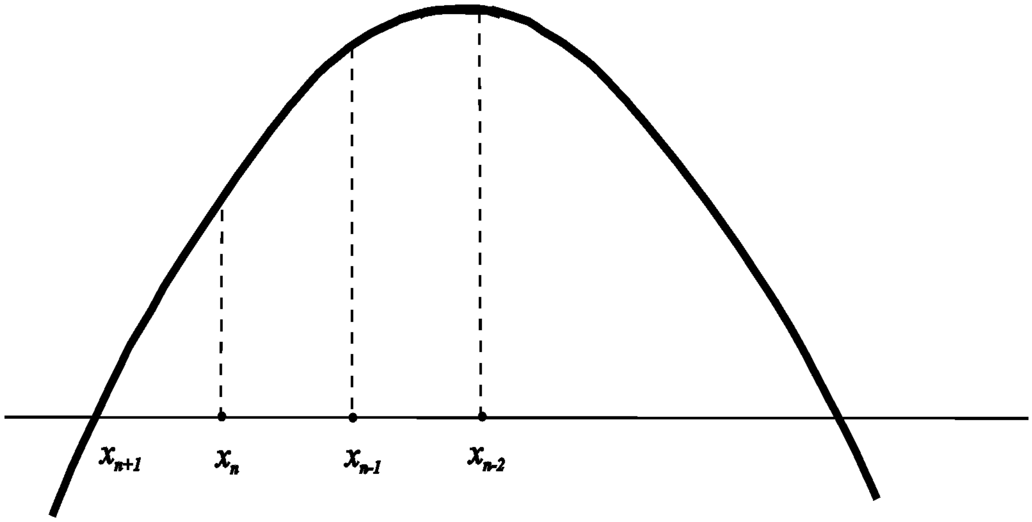


Рис. 18

$x_{n+1} = c$, где

$$
\begin{aligned}
c &= x_n - af_n^2 - bf_n, \\
b &= -a(f_n + f_{n-1}) + \frac{x_n - x_{n-1}}{f_n - f_{n-1}}, \\
a &= \frac{(x_n - x_{n-1})(f_{n-1} - f_{n-2}) - (x_{n-1} - x_{n-2})(f_n - f_{n-1})}{(f_n - f_{n-1})(f_{n-1} - f_{n-2})(f_n - f_{n-2})}
\end{aligned}
$$

Рассмотренный метод требует меньших затрат, чем метод парабол, поскольку в нем используется только вещественная арифметика и не требуется извлечения квадратных корней. Более того, его скорость сходимости остается той же, равной приблизительно 1.84. Однако метод обратной квадратичной интерполяции требует задания хороших начальных приближений, что снижает его практическое значение по сравнению с методом парабол.

17. Еще раз о комбинированных методах

В рассмотренных выше методах мы не учитывали локальные свойства функций на выделенных отрезках, содержащих корни. Однако ясно, что если мы будем их учитывать, то можно динамически переключаться с одного метода на другой в зависимости от поведения функции, повышая тем самым скорость сходимости.

Рассмотрим вначале комбинации методов хорд и половинного деления. На рис. 20 показан случай, когда метод хорд дает худшее приближение по сравнению с методом половинного деления.

Может быть предложен следующий алгоритм. На первых шагах применяется метод половинного деления $c^n = (a^n + b^n)/2$ и после вычисления решается вопрос, лежит ли точка $(c^n; f(c^n))$ на прямой, проходящей через точки $(a^n; f(a^n))$ и $(b^n; f(b^n))$. Из аналитической геометрии известно, что условием принадлежности трех точек одной прямой является равенство нулю следующего определителя:

$$
\begin{vmatrix} a^n & f(a^n) & 1 \\ c^n & f(c^n) & 1 \\ b^n & f(b^n) & 1 \end{vmatrix} = 0.
$$

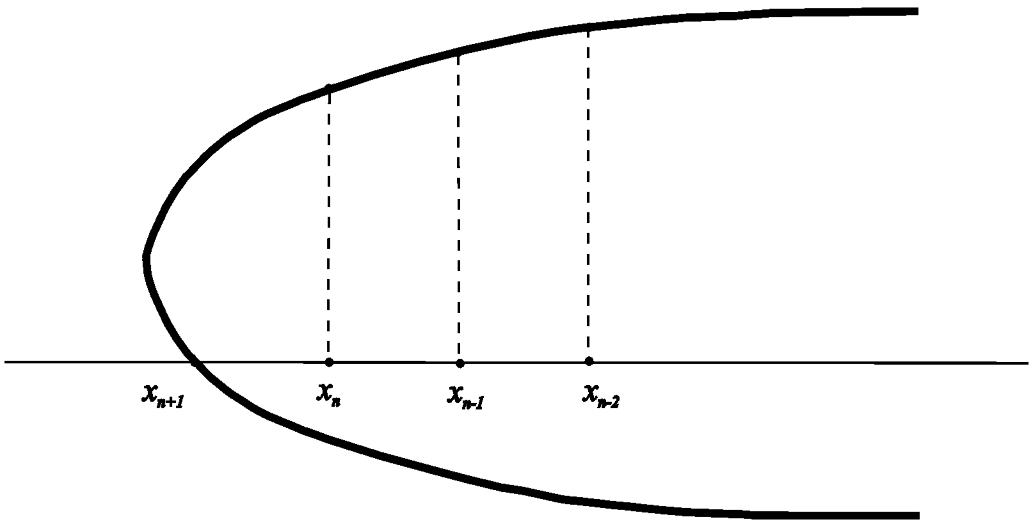


Рис. 19

Раскрыв определитель, будем иметь

$$c^n f(b^n) - b^n f(c^n) - a^n f(b^n) + b^n f(a^n) + a^n f(c^n) - c^n f(a^n) = 0.$$

После приведения подобных членов имеем

$$(b^n - c^n) f(a^n) - (b^n - a^n) f(c^n) + (c^n - a^n) f(b^n) = 0.$$

Поскольку $c^n = (a^n + b^n)/2$, то

$$\frac{b^n - a^n}{2} f(a^n) - \frac{2(b^n - a^n)}{2} f(c^n) + \frac{b^n - a^n}{2} f(b^n) = 0.$$

т.е.

$$f(a^n) - 2f(c^n) + f(b^n) = 0,$$

если все рассматриваемые точки принадлежат одной прямой. Естественно считать условием принадлежности этих точек одной прямой выполнение неравенства

$$|f(a^n) - 2f(c^n) + f(b^n)| < \gamma\, |f(b^n) - f(a^n)|,$$

где параметр γ подбирается экспериментально.

В случае положительного ответа, начиная с этого момента в качестве c^{n+1} берут приближение по методу хорд, а в противном случае применяют метод половинного деления и рассматривают вопрос о принадлежности новых трех точек одной прямой.

Второй комбинированный метод, получивший название *метода Брента*, является двусторонним и представляет собой объединение методов обратной квадратичной интерполяции, секущих и половинного деления. На каждой итерации всегда вначале делается попытка применить обратную квадратичную интерполяцию. Если полученное приближение “разумно”, т.е. оно остается внутри текущего интервала, на котором функция меняет знак, и не слишком близка к его концам, то итерация считается завершенной. В противном случае применяется метод секущих и полученное приближение также проверяется на “разумность”. Если оно также неудовлетворительно, то применяется метод половинного деления. Метод Брента относится к числу наиболее эффективных практических методов.

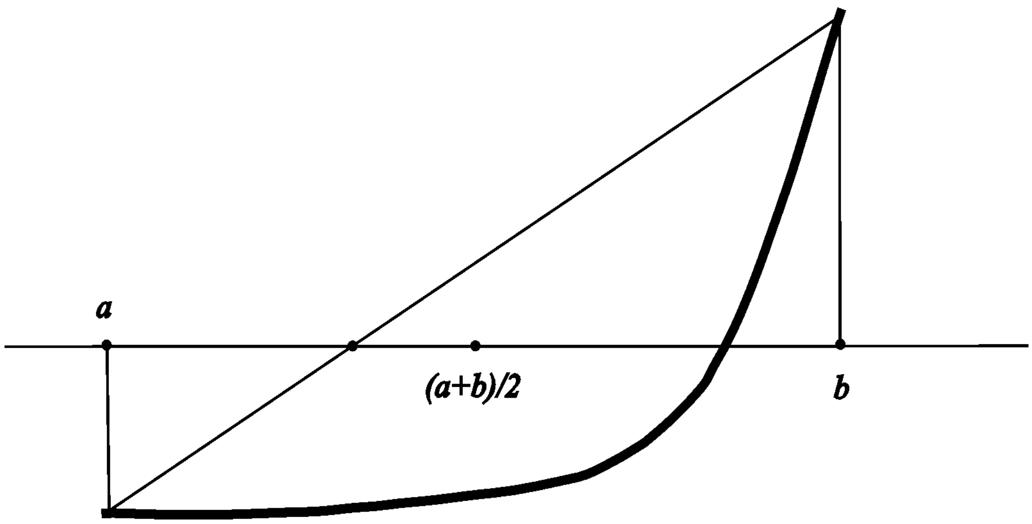


Рис. 20

18. Вычисление комплексных, кратных и нескольких корней

Для того чтобы вычислить комплексный корень, если функция $f(x)$ вещественная при вещественном аргументе, в методах простых итераций, Ньютона или секущих необходимо задать комплексное начальное приближение. В методе парабол сходимость к комплексному корню возможна и при вещественных начальных приближениях.

Для вычисления корней нечетной кратности применимы все рассмотренные методы, поскольку в этом случае возможно выделить отрезки, на которых функция меняет знак. Если кратность корня четна, то применяют методы Ньютона или секущих, а также метод парабол.

Если кратность корня заранее неизвестна или ищутся несколько корней, то прибегают к следующему способу.

Если $\bar{x}_1$ является простым корнем и для функции $f(x)$ на рассматриваемом отрезке выполняется неравенство $|f(x') - f(x'')| \leq K\delta$ для любой пары точек отрезка, удовлетворяющих условию $|x' - x''| \leq \delta$, где K — некоторая константа (*константа Липшица*), то вспомогательная функция

$$g_1(x) = \frac{f(x)}{x - \bar{x}_1}$$

непрерывна, причем все нули функций $f(x)$ и $g(x)$ совпадают, за исключением $\bar{x}_1$, поскольку $g_1(\bar{x}_1) \neq 0$. Если $\bar{x}_1$ является кратным корнем, то он будет нулем функции $g_1(x)$, но кратности на единицу меньше, а остальные нули обеих функций будут одинаковыми. Отсюда следует, что найденный корень всегда можно удалить, перейдя к функции $g_1(x)$. Тогда поиск остальных нулей функции $f(x)$ сводится к поиску нулей функции $g_1(x)$. Если найден какой-либо нуль $\bar{x}_2$ функции $g_1(x)$, то и его можно удалить, вводя другую вспомогательную функцию

$$g_2(x) = \frac{g_1(x)}{x - \bar{x}_2} = \frac{f(x)}{(x - \bar{x}_1)(x - \bar{x}_2)}.$$

Повторяя этот процесс, мы находим требуемое количество корней.

Однако в действительности мы находим лишь приближенное значение корня $\tilde{x}_1 \approx \bar{x}_1$ и строим функцию не $g_1(x)$, а функцию $\tilde{g}_1(x) = \dfrac{f(x)}{x - \tilde{x}_1}$, которая имеет нуль в точке $\bar{x}_1$ и полюс в близкой к ней точке $\tilde{x}_1$. Чтобы это не сказывалось при нахождении следующих корней, надо вычислять каждый корень с высокой точностью, особенно если он кратный или вблизи него расположен другой корень.

Кроме того, для любого метода окончательные итерации вблизи определяемого корня рекомендуется делать не по функциям $g_i(x)$, а по исходной функции $f(x)$. При этом последние итерации, вычисленные по функции $g_i(x)$, используются в качестве нулевых приближений. Это особенно важно при вычислении многих корней, поскольку чем больше корней удалено, тем меньше нули вспомогательной функции $\dfrac{f(x)}{\prod\limits_i (x-\tilde{x}_i)}$ соответствуют остальным нулям функции $f(x)$.

Для нахождения n нулей функции, когда нет информации относительно начальных приближений к ним или же начальные приближения грубые, применяют метод парабол. Если начальные приближения к нулям достаточно хорошие, то разумно применение методов секущих или Ньютона, когда возможны нули четной кратности, или же двустороннего метода Брента в противном случае.

19. Корни многочленов

Для нахождения корней многочленов рекомендуется метод парабол. Хотя его сходимость при произвольном начальном приближении и не доказана, но на практике этот метод быстро сходится к какому-нибудь корню. Поскольку для многочлена частное $\dfrac{f(x)}{x-\bar{x}}$ также многочлен, то последовательно удаляя найденные корни можно найти все корни исходного многочлена.

Следует учитывать возможную сильную неустойчивость корней многочленов как функций от их коэффициентов. Наиболее чувствительными могут быть наибольшие по модулю корни, особенно к погрешностям коэффициентов при старших степенях. Кратные или близкие корни могут быть слабо устойчивыми и при небольших степенях многочленов. Это означает, что необходимо предельно точно вычислять значение коэффициентов.

Если многочлен имеет высокую степень, то возникают дополнительные трудности, связанные с тем, что многочлен быстро возрастает при увеличении аргумента. Поэтому при численных расчетах необходимо учитывать возможности переполнений. Обычно вводят *масштабные множители*, величина которых связана с диапазоном изменений аргумента.

При удалении корней вносится погрешность округлений в коэффициенты, а это влияет на точность вычисления следующих корней. Чтобы уменьшить это влияние, на практике начинают с удаления меньших по модулю корней, поскольку если начать с удаления больших корней, то точность может упасть катастрофически. Поэтому за начальное приближение рекомендуется взять $x_0 = -1$, $x_1 = 1$, $x_0 = 0$. Тогда итерации обычно сходятся к наименьшему по модулю корню. После его удаления по тем же начальным приближениям ищут следующий корень и т.д. При такой организации вычислений потеря точности будет небольшой.

Литература

1. *Арушанян И.О., Чижонков Е.В.* Материалы семинарских занятий по курсу “Методы вычислений” / под ред. Арушаняна О.Б. — М.: Изд-во ЦПИ при механико-математическом факультете МГУ, 1999.

2. *Бахвалов Н.С.* Численные методы. М.: Наука, 1975.

3. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.* Численные методы. М.: Наука, 1987.

4. Библиотека НИВЦ МГУ решения типовых задач численного анализа (http://www.srcc.msu.su/num_anal).

5. *Демидович Б.П., Марон И.А.* Основы вычислительной математики. М.: Физматгиз, 1963.

6. *Калиткин Н.Н.* Численные методы. М.: Наука, 1978.

7. *Каханер Д., Моулер К., Нэш С.* Численные методы и программное обеспечение. М.: Мир, 1998.

8. *Крылов В.И., Бобков В.В., Монастырный П.И.* Вычислительные методы. М.: Наука, 1977.