

Глава 9 МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ И СИСТЕМ

В решении нелинейных уравнений

$$f(x)=0 \quad (2.1)$$

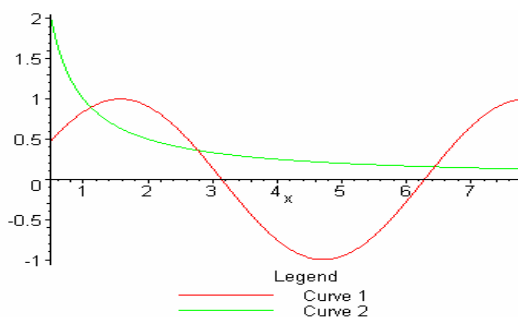
можно выделить два этапа. Первый этап связан с исследованием количества, характера расположения корней и нахождением их грубого приближения. Уравнение (2.1) может вообще не иметь решений, а может встретиться ситуация (например, уравнение $x=ig(x)$), когда число корней бесконечно. Этот этап формализуется лишь частично и чаще относится к области математического искусства.

На втором этапе предполагается, что один из корней локализован, и требуется уточнить его значение. Иногда из нелинейных уравнений общего вида (2.1) выделяют алгебраические, специфические свойства которых можно использовать при решении.

При локализации корней на первом этапе часто строят таблицу значений функции $f(x)$ и находят участок, где она меняет знак. Это свидетельствует о нечетном числе корней и при малой длине участка можно надеяться, что корень будет лишь один. С другой стороны, уравнение (2.1) иногда можно представить в виде

$$\varphi(x)=\psi(x),$$

когда графики функций $\varphi(x)$ и $\psi(x)$ легко строятся, а абсциссы точек их пересечения будут корнями уравнения (2.1). Например, уравнение $x \cdot \sin(x)=1$ можно переписать так: $\sin(x)=1/x$. Вот иллюстрация при $x \in [0.5, 2.5\pi]$.



§ 1 Уточнение корней одного уравнения

Остановимся лишь на вещественных корнях уравнения (2.1), считая, что функция $f(x)$ нужное для выбранного метода число раз непрерывно дифференцируема и установлен промежуток $[a, b]$, где находится единственный корень. При этом $f(a) f(b) < 0$. Тогда наиболее простым и абсолютно надежным способом является метод бисекции (или метод дихотомии или метод половинного деления). Его схему представим следующим образом:

Шаг 1. Вычислить $f(a)$ и $f(b)$.

Шаг 2. Положить $c = (a+b)/2$ и вычислить $f(c)$.

Шаг 3. Если $\text{sign}(f(a)) = \text{sign}(f(c))$, заменить a на c . В противном случае заменить b на c .

Шаг 4. Если $|b-a| > \varepsilon$, то перейти к шагу 2. В противном случае закончить вычисления.

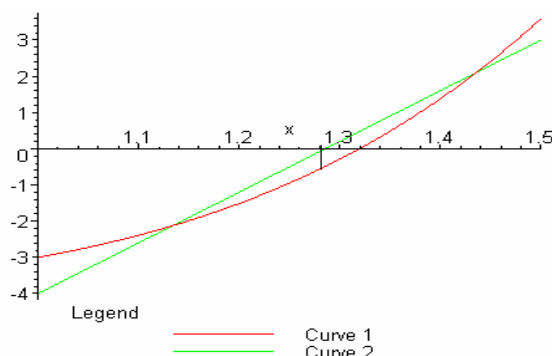
Одна итерация алгоритма позволяет гарантированно сократить исходный промежуток в два раза независимо от вида функции. Для уточнения трех десятичных цифр требуется не менее 10 делений промежутка ($2^{10} = 1024 \approx 10^3$).

Вопрос: "Это свойство метода является его достоинством или недостатком?"

Обратим внимание на тот факт, что количество чисел, представимых в ЭВМ, конечно, и может не найтись числа, которое обеспечит абсолютно точное равенство в уравнении (2.1). Поэтому и ищется малый промежуток, ограниченный снизу дискретностью представления чисел в ЭВМ, на котором $f(x)$ меняет знак.

Другой алгоритм, называемый методом секущих, может быть получен построением интерполяционного полинома Лагранжа первой степени для $f(x)$ и нахождением нуля этого полинома в качестве очередного приближения к корню уравнения (2.1)

$$L_1(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b) = 0, \quad c = a - \frac{b-a}{f(b)-f(a)} f(a)$$



Новый промежуток будет $[c, b]$ или $[a, c]$ в зависимости от знака $f(x)$ в точке c . Скорость сходимости метода секущих определяется неравенством $|x^* - x_{k+1}| \leq |x^* - x_k|^{1.618}$. Следует отметить, что замедление сходимости этого алгоритма часто наблюдается, когда очередное приближение получается слишком близко к одному из концов промежутка.

Если функция вычислена более чем в двух точках, то эта информация может быть использована в дальнейшем. Так, в методе обратной квадратичной интерполяции строится интерполяционный полином второй степени по трем точкам x_k, x_{k-1}, x_{k-2} для обратной функции с выполнением условий $x_i = g(f_i)$ $i = k, k-1, k-2$. В качестве следующего приближения берется $x_{k+1} = g(0)$. Одна из предыдущих точек удаляется. Важно, чтобы три значения f_i были бы различными, тогда исключается деление на ноль, и сходимость метода определяется неравенством $|x^* - x_{k+1}| \leq |x^* - x_k|^{1.839}$, где x^* - точное решение (2.1).

Сочетание методов бисекции и обратной квадратичной интерполяции реализовано в процедуре-функции ZEROIN, написанной на Фортране [5]. Основным алгоритмом является метод обратной квадратичной интерполяции (если x_k, x_{k-1}, x_{k-2} не являются различными, используется метод секущих). Если очередное приближение получается слишком близким к краям промежутка, то осуществляется переключение на метод бисекции. Обращение к программе имеет вид

$$R = \text{ZEROIN}(A, B, F, \text{EPS}),$$

где A и B - концы интервала, где ищется корень; F - имя процедуры-функции, имеющей лишь один аргумент, для которого вычисляется $f(x)$, EPS - граница погрешности, допустимой в результате.

Еще одним методом для решения (2.1) является метод последовательных приближений, для построения которого эквивалентными преобразованиями приведем (2.1) к виду

$$x = g(x) \quad (2.2)$$

где корень x^* уравнения (2.2) является корнем и (2.1). Введем также обозначения $x^* = x_n + \varepsilon_n$, где x_n - n -ое приближение к x^* , ε_n - погрешность. Вместо уравнения (2.2) предлагается решать разностное уравнение

$$x_{n+1} = g(x_n) \quad (2.3)$$

пошаговым методом. Для оценки сходимости запишем равенство (2.2) в точке x^* и вычтем из него равенство (2.3):

$$\varepsilon_{n+1} = x^* - x_n = g(x^*) - g(x_n)$$

Раскладывая $g(x^*)$ в ряд по степеням ε_n и ограничиваясь в остаточном члене первой производной, получаем уравнение погрешности

$$\varepsilon_{n+1} = g(x_n + \varepsilon_n) - g(x_n) = \varepsilon_n g'(\xi).$$

Отсюда непосредственно следует, что для убывания погрешности необходимо потребовать выполнение условия

$$g'(\xi) < 1. \quad (2.4)$$

Искусство пользователя заключается в приведении уравнения (2.1) к виду (2.2) так, чтобы имело место неравенство (2.4). При этом, чем меньше по модулю значение производной, тем быстрее достигается желаемая точность.

Высокой скоростью сходимости в ряде случаев обладает метод касательных или метод Ньютона. Подставляя в уравнение (2.1) его корень x^* , получаем

$$f(x^*) = f(x_n + \varepsilon_n) = f(x_n) + \varepsilon_n f'(x_n) + \varepsilon_n^2 f''(\xi)/2. \quad (2.5)$$

Если пренебречь последним слагаемым в (2.5), то для ε_n , имеем $\varepsilon_n = -f(x_n)/f'(x_n)$, и рабочая формула метода Ньютона приобретает вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2.6)$$

В отличие от методов бисекции, секущих и обратной квадратичной интерполяции сходимость (2.6) обеспечивается далеко не всегда. Необходимым условием является не только существование ненулевой производной $f'(x)$ в точках x_n , но и ее знакопостоянство. Если тем не менее методу Ньютона обеспечено хорошее начальное приближение, то в дальнейшем убывание погрешности носит квадратичный характер. Для доказательства этого факта из очевидного уравнения $x^* = x^*$ вычтем уравнение (2.6):

$$\varepsilon_{n+1} = \varepsilon_n + \frac{f(x_n)}{f'(x_n)} = \frac{f(x_n) + \varepsilon_n f'(x_n)}{f'(x_n)} \quad (2.7)$$

Упрощая числитель (2.7) с помощью равенства (2.5), получаем

$$\varepsilon_{n+1} = -\frac{f''(\xi)}{2f'(x_n)} \varepsilon_n^2, \quad |\varepsilon_{n+1}| < c \varepsilon_n^2, \quad \left| \frac{f''(\xi)}{2f'(x_n)} \right| < c.$$

Метод Ньютона называют еще методом касательных, так как формула (2.6) представляет собой уравнение касательной

$$Q_1(x) = f(x_n) + f'(x_n)(x - x_n),$$

нулевое значение которой отвечает следующему приближению x_{n+1} .

Интересно заметить, что для уменьшения ошибки до половины наименьшего возможного значения в формате double (8 байтов) требуется 6 итераций, в то время как метод бисекции потребовал бы 56 итераций.

Для расширения области сходимости можно использовать метод Ньютона с регулировкой шага

$$x_{n+1} = x_n - \alpha_n \frac{f(x_n)}{f'(x_n)}, \quad 0 < \alpha_n \leq 1.$$

Существует теорема, формулирующая условие выбора α_n . На практике же вдали от корня α_n выбирают примерно 1/3, а вблизи - около единицы, приближаясь к (2.6). Широкое распространение получил и модифицированный метод Ньютона с постоянным значением производной:

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

Сходимость в этом случае несколько замедляется, но зато уменьшается трудоемкость отдельной итерации, не требующей теперь вычисления производной.

§ 2 Метод Ньютона для систем уравнений

Решение систем нелинейных уравнений доставляет очень большие трудности, поскольку нет универсальных алгоритмов решения этих задач, особенно для больших m .

Достоинством метода Ньютона по сравнению со многими алгоритмами предыдущего параграфа является то, что он обобщается на системы уравнений. С этой целью обратимся к уравнению (2.1), полагая x и f векторами: $x = (x^{(1)}, x^{(2)}, \dots, x^{(m)})^T$, $f = (f^{(1)}, f^{(2)}, \dots, f^{(m)})^T$.

Формула, являющаяся аналогом (2.6), может быть получена таким же образом, как и для скалярного случая, только представляет собой вектор и разложение в ряд необходимо проводить для функции многих переменных. В настоящем разделе используем другой подход, основанный на лемме Адамара.

Рассмотрим вектор-функцию $f(x + t\varepsilon_n)$ скалярного аргумента t . При этом величина ε_n считается не зависящей от t . Вычислим производную этой функции по t и проинтегрируем результат по t от 0 до 1.

$$\int_0^1 \frac{d}{dt}(f(x + t\varepsilon_n)) dt = \int_0^1 \frac{\partial f(x + t\varepsilon_n)}{\partial x} dt \varepsilon_n,$$

$$f(x + \varepsilon_n) - f(x) = \int_0^1 \frac{\partial f(x + t\varepsilon_n)}{\partial x} dt \varepsilon_n.$$

Это и есть лемма Адамара.

Полагая $x = x_n$ и принимая $x + \varepsilon_n$ за очередное приближение к искомому корню уравнения (2.1), т.е. считая $f(x + \varepsilon_n)$ равной нулю, имеем

$$x_{n+1} - x_n = \varepsilon_n = - \left(\int_0^1 \frac{\partial f(x + t\varepsilon_n)}{\partial x} dt \right)^{-1} f(x_n).$$

Это и есть метод Ньютона.

Вычисляя интеграл в последней формуле по квадратурной формуле левых прямоугольников, получаем метод Ньютона для систем уравнений в традиционной форме

$$x_{n+1} = x_n - \left(\frac{\partial f(x_n)}{\partial x_n} \right)^{-1} f(x_n)$$

или

$$\frac{\partial f(x_n)}{\partial x_n} (x_{n+1} - x_n) = -f(x_n) \quad (2.9)$$

Представление метода в виде (2.9) позволяет уменьшить вычислительные затраты, поскольку не требует обращения матрицы Якоби $\left(\frac{\partial f(x_n)}{\partial x_n} \right)$, а сводится к решению линейной алгебраической системы на каждом шаге итерационного процесса. Как и в скалярном случае, в достаточно малой окрестности корня итерации сходятся и скорость сходимости квадратичная. Значительно уменьшается объем вычислений в модифицированном варианте метода Ньютона, когда матрица Якоби вычисляется однократно, раскладывается в произведение треугольных матриц программой DECOMP, а затем для получения очередного приближения используется только программа SOLVE.

Для решения систем уравнений может быть использован и метод последовательных приближений. Формула (2.3) сохраняет прежний вид, только x и $g(x)$ являются векторами.

Достаточным условием сходимости является выполнение неравенства $\left\| \frac{\partial g}{\partial x} \right\| \leq 1$, где $\left(\frac{\partial g}{\partial x} \right)$ - матрица Якоби.

§ 3 Методы минимальных невязок Ракитского

Решение систем нелинейных уравнений может быть построено на идее минимизации некоторой функции подобно тому, как для линейных систем строился метод минимальных невязок (1.13)-(1.15). Так, для (2.1) Ю.В.Ракитским были предложены однопараметрический и трехпараметрический методы минимальных невязок. Начнем с однопараметрического метода, полагая, что $f(x)$ - дважды дифференцируемая вектор-функция векторного аргумента, отделение корней произведено и начальное приближение x_0 находится в некоторой r - окрестности корня: $\|x^* - x_0\| < r$.

В первой итерации новое приближение находится по формуле

$$x_1 = x_0 + \alpha_0 f(x_0).$$

Учитывая (2.1), построим невязку в виде $R(\alpha_0) = (f(x_1), f(x_1))$ и потребуем ее минимальности по параметру α_0 . Допустим, что при достаточно малых $|\alpha_0| < \delta$ невязку $R(\alpha_0)$ можно разложить в степенной ряд по степеням α_0 .

$$R(\alpha_0) = R(0) + \alpha_0 R'(0) + \alpha_0^2 / 2 R''(0) + o(\alpha_0^3).$$

Минимум обеспечивается условием $R'(\alpha_0) = 0$, которое при пренебрежении в разложении слагаемыми с α_0 выше второй степени приводит к заданию α_0 в соответствии с формулой

$$\alpha_0 = -R'(0)/R''(0). \quad (2.10)$$

Если в частном случае система (2.1) является линейной, то $f(x)$ является вектором невязки и формула переходит в выражение (1.11) для метода минимальных невязок для систем (1.1). В нелинейном же случае значения $R'(0)$ и $R''(0)$ вычисляются по формулам численного дифференцирования. Выберем некоторое значение $|h_0| < \delta$ и введем обозначения: $R(0) = R_0$, $R(h_0) = R_+$, $R(-h_0) = R_-$. Тогда

$$R'(0) = \frac{R_+ - R_-}{2h_0}, \quad R''(0) = \frac{R_+ - 2R_0 + R_-}{h_0^2}$$

и формула для α_0 приобретает вид

$$\alpha_0 = -\frac{R'(0)}{R''(0)} \cong -\frac{h_0}{2} \frac{R_+ - R_-}{R_+ - 2R_0 + R_-}.$$

Последующие итерации выполняются аналогично

$$x_{k+1} = x_k + \alpha_k f(x_k),$$

$$\alpha_k = -\frac{R'(0)}{R''(0)} \cong -\frac{h_k}{2} \frac{R(h_k) - R(-h_k)}{R(h_k) - 2R(0) + R(-h_k)}.$$

На каждом шаге вектор-функция $f(x)$ вычисляется три раза, а условие успешного выполнения итераций выглядит следующим образом:

$$\|f(x_n)\| \leq \|f(x_{n-1})\|.$$

Теперь обратимся к трехпараметрическому методу. По формулам однопараметрического метода для выбранного начального приближения x_0 находим

$$x_1^0 = x_0 + \alpha_0 f(x_0).$$

Далее вычисляем $f(x_1^0)$ и строим функцию $P_0 = f(x_0) + \beta_0 f(x_1^0)$ с параметром β_0 , который находим из условия ортогональности P_0 и $f(x_0)$:

$$\beta_0 = -\frac{(f(x_0), f(x_0))}{(f(x_0), f(x_1^0))}.$$

Затем строим x_1 , по формуле $x_1 = x_0 + \alpha_0 f(x_1^0) + \gamma_0 P_0$ и γ_0 находим из условия минимальности скалярного произведения:

$$S_0(\gamma_0) = (f(x_1), f(x_1)).$$

Как и в однопараметрическом методе, разлагаем $S_0(\gamma_0)$ в ряд и по условию минимальности приходим к равенству $\gamma_0 = -S'(0)/S''(0)$. Аналогично воспользуемся формулами численного дифференцирования: $(|q_0| < \delta$

$$\gamma_0 = -\frac{S'(0)}{S''(0)} \cong -\frac{q_0}{2} \frac{S_+ - S_-}{S_+ - 2S_0 + S_-},$$

где q_0, S_+, S_- имеют смысл аналогичный таковому в однопараметрическом методе.

Последующие итерации выполняются в соответствии с равенствами

$$\begin{aligned} \mathbf{x}_{k+1}^0 &= \mathbf{x}_k + \alpha_k \mathbf{f}(\mathbf{x}_k), \quad \mathbf{R}_k(\alpha_k) = (\mathbf{f}(\mathbf{x}_{k+1}^0), \mathbf{f}(\mathbf{x}_{k+1}^0)), \\ \alpha_k &= -\frac{q_k}{2} \frac{R(q_k) - R(-q_k)}{R(q_k) - 2R(0) + R(-q_k)} \end{aligned}$$

Так завершается прогноз по однопараметрическому методу. Далее вычисления проводятся по формулам:

$$\begin{aligned} \mathbf{P}_k &= \mathbf{f}(\mathbf{x}_k) + \beta_k \mathbf{f}(\mathbf{x}_{k+1}^0), \quad \beta_k = -\frac{(\mathbf{f}(\mathbf{x}_k), \mathbf{f}(\mathbf{x}_k))}{(\mathbf{f}(\mathbf{x}_k), \mathbf{f}(\mathbf{x}_{k+1}^0))}, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{f}(\mathbf{x}_{k+1}^0) + \gamma_k \mathbf{P}_k, \\ \mathbf{S}_k(\gamma_k) &= (\mathbf{f}(\mathbf{x}_{k+1}), \mathbf{f}(\mathbf{x}_{k+1})), \quad \gamma_k = -\frac{q_k}{2} \frac{S_k(q_k) - S_k(-q_k)}{S_k(q_k) - 2S_k(0) + S_k(-q_k)}, \end{aligned}$$

В трехпараметрическом методе шесть вычислений значений вектор-функции $\mathbf{f}(\mathbf{x})$ в каждой итерации: $\mathbf{f}(\mathbf{x}_k)$, $\mathbf{f}(\mathbf{x}_k + q_k \mathbf{f}(\mathbf{x}_k))$, $\mathbf{f}(\mathbf{x}_k - q_k \mathbf{f}(\mathbf{x}_k))$, $\mathbf{f}(\mathbf{x}_{k+1}^0)$, $\mathbf{f}(\mathbf{x}_{k+1}^0 + q_k \mathbf{P}_k)$, $\mathbf{f}(\mathbf{x}_{k+1}^0 - q_k \mathbf{P}_k)$.

Условия успешного выполнения итераций: $\|\mathbf{f}(\mathbf{x}_{k+1})\| \leq \|\mathbf{f}(\mathbf{x}_{k+1}^0)\| \leq \|\mathbf{f}(\mathbf{x}_k)\|$. Вопросы сходимости этих методов еще в полной мере изучены.