1.10 Численные методы решения нелинейных алгебраических уравнений и систем. Итерационные методы, линеаризация по Ньютону, методы спуска.

Термины:

Корень функции – это такое значение ее аргумента, при котором функция равна нулю.

Корень алгебраического уравнения – это такое значение его неизвестного, при котором уравнение становится тождеством.

Корень системы алгебраических *уравнений* – это такое значение ее неизвестных, при которых уравнения, входящие в систему, становятся тождествами.

Решение алгебраического уравнения или системы – нахождение его (ее) корней

Основные особенности *численного* поиска корней нелинейных алгебраических уравнений и систем такие:

Определяется *один* корень из многих существующих. Какой именно корень будет найден – зависит главным образом от первого приближения к решению (guess values), который должен задать сам пользователь. Можно задать несколько приближений вектором

Ищется не корень уравнения или системы, а значение неизвестного (неизвестных), при котором отклонение правых и левых частей уравнений (*неувязка*) не превышает наперед заданного числа.

Итерационный метод(Метод простой итерации):

Метод простой итерации

В основе метода заложено понятие сжимающего отображения. Определим терминологию:

Говорят, что функция ϕ осуществляет ***сжимающее отображение*** на $[a,\ b]$, если

$$\forall x \in [a,\ b] : \phi(x) \in [a,\ b]$$

$$\exists \alpha < 1 : \forall x_1, x_2 \in [a,\ b] \quad ||\phi(x_1) - \phi(x_2)|| \le \alpha ||x_1 - x_2||$$

Тогда основная теорема будет выглядеть так:

Теорема Банаха (принцип сжимающих отображений).

Если ϕ — сжимающее отображение на $[a,\ b]$, то:

у $x = \phi(x) \quad \exists! x^* \in [a,\ b]$ — корень;

итерационная последовательность $x_{i+1} = \phi(x_i)$сходится к этому корню;

для очередного члена x_nсправедливо $||x_n - x^*|| \le \dfrac{\alpha^n ||x_1 - x_0||}{1-\alpha}$.

Поясним смысл параметра α. Согласно теореме Лагранжа имеем:

$$\phi(x) \in C^1[a,\ b]. \quad \forall x_1, x_2 \in (a,\ b), \quad x_1 < x_2 \quad \exists \xi \in (x_1,\ x_2): \quad \phi'(\xi)(x_2 - x_1) =$$

Отсюда следует, что $\alpha \approx |\phi'(\xi)|$. Таким образом, для сходимости метода достаточно, чтобы $\forall x \in [a,\ b] \quad |\phi'(x)| \le 1.$

Применительно к СЛАУ

Рассмотрим систему:

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n & = & b_1 \\ & \ldots & \\ a_{n1}x_1 + \ldots + a_{nn}x_n & = & b_n \end{cases}$$

Для неё итерационное вычисление будет выглядеть так:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}^{i+1} = \begin{pmatrix} a_{11}+1 & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22}+1 & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn}+1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}^{i} - \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Сходимость методу будет осуществлять $\begin{vmatrix} a_{11}+1 & \ldots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn}+1 \end{vmatrix} < 1$

Алгоритм

1. Условие $f(x) = 0$преобразуется к виду $x = \phi(x)$, где $\phi(x)$— сжимающая
2. Задаётся начальное приближение и точность $x_0, \quad \varepsilon, \quad i = 0$
3. Вычисляется очередная итерация $x_{i+1} = \phi(x_i)$
4. Если $||x_{i+1} - x_i|| > \varepsilon$, то $i = i + 1$ и возврат к шагу 3.
5. Иначе $x = x_{i+1}$и останов.

Метод Ньютона (метод касательных)

В одномерном варианте еще есть метод хорд.

Одномерный случай

Для того, чтобы решить уравнение $f(x) = 0$, пользуясь методом простой итерации, необходимо привести его к виду $x = \phi(x)$, где ϕ— сжимающее отображение. Чтобы отображение было наиболее эффективно, необходимо, чтобы в точке очередной итерации x * выполнялось $\phi'(x^*) = 0$.

Будем искать решение данного уравнения в виде $\phi(x) = x + \alpha(x)f(x)$, тогда:

$$\phi'(x^*) = 1 + \alpha'(x^*)f(x^*) + \alpha(x^*)f'(x^*) = 0$$

Воспользуемся тем, что $f(x) = 0$, и получим окончательную формулу для $\alpha(x)$:

$$\alpha(x) = -\frac{1}{f'(x)}$$

С учётом этого сжимающая функция примет вид:

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

Тогда алгоритм нахождения численного решения уравнения $f(x) = 0$ сводится к итерационной процедуре вычисления:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Многомерный случай

Обобщим полученный результат на многомерный случай.

Выбирая некоторое начальное приближение $\vec{x}^{[0]}$, находят последовательные приближения $\vec{x}^{[j+1]}$путем решения систем уравнений:

$$f_i + \sum_{k=1}^{n} \frac{\partial f_i}{\partial x_k}(x_k^{[j+1]} - x_k^{[j]}) = 0, \quad i = 1, 2, \dots, n$$
,

где $x^{[j]} = \left(x_1^{[j]} \dots x_k^{[j]} \dots x_n^{[j]}\right), \quad j = 0, 1, 2, \dots$.

Метод спуска(градиентный спуск, наискорейший спуск)

Градиентный спуск — метод нахождения локального минимума (максимума) функции с помощью движения вдоль антиградиента (градиента). Для минимизации функции в направлении градиента используются методы одномерной оптимизации, например, метод золотого сечения. Также можно искать не наилучшую точку в направлении градиента, а какую-либо лучше текущей.

Если модифицировать исходную функцию, добавив к ней «штраф» возрастающий при отдалении значения функции от нуля, то метод нахождения локального минимума превратится в метод нахождения корня.

Пусть целевая функция имеет вид:

$$f(\vec{x}):\ \mathbb{X} \to \mathbb{R}.$$

И задача оптимизации задана следующим образом:

$$f(\vec{x}) \to \min_{\vec{x}\in\mathbb{X}}$$

Основная идея метода заключается в том, чтобы идти в направлении наискорейшего спуска, а это направление задаётся антиградиентом $-\nabla F$:

$$\overrightarrow{x}^{[j+1]} = \overrightarrow{x}^{[j]} - \lambda^{[j]}\nabla F(\overrightarrow{x}^{[j]})$$

где $\lambda^{[j]}$ выбирается

- постоянной, в этом случае метод может расходиться;
- дробным шагом, т.е. длина шага в процессе спуска делится на некое число;
- наискорейшим спуском: $\lambda^{[j]} = \mathrm{argmin}_\lambda\, F(\vec{x}^{[j]} - \lambda^{[j]}\nabla F(\vec{x}^{[j]}))$

Алгоритм

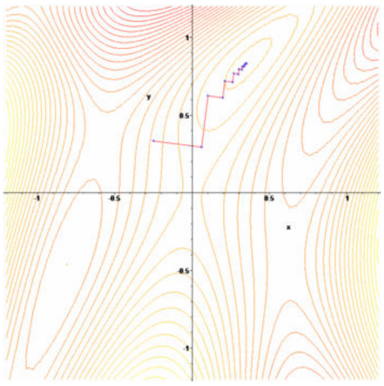
1. Задают начальное приближение и точность расчёта $\vec{x}^0, \quad \epsilon$
2. Рассчитывают $\overrightarrow{x}^{[j+1]} = \overrightarrow{x}^{[j]} - \lambda^{[j]}\nabla F(\overrightarrow{x}^{[j]})$, где $\lambda^{[j]} = \mathrm{argmin}_\lambda\, F(\vec{x}^{[j]} - \lambda^{[j]}\nabla F(\vec{x}^{[j]}))$
3. Проверяют условие останова:
   - Если $|\vec{x}^{[j+1]} - \vec{x}^{[j]}| > \epsilon$, то $j = j + 1$ и переход к шагу 2.
   - Иначе $\vec{x} = \vec{x}^{[j+1]}$ и останов.

Пример

Применим градиентный метод к функции

$$F(x,y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y).$$

Тогда последовательные приближения будут выглядеть так:



y
x
1
0.5
-0.5
-1
-1
-0.5
0.5
1