

Задание №6

Многомерный метод Ньютона

Описание алгоритма

Метод предназначен для решения системы нелинейных алгебраических уравнений вида $F(X) = 0$ или

$$\begin{cases} F_1(X_1, X_2, \dots, X_n) = 0 \\ \dots \\ F_n(X_1, X_2, \dots, X_n) = 0 \end{cases}$$

Выбираем начальное приближение $X^{(0)}$ и решаем систему итерациями, на каждом шаге j получая решение системы линейных алгебраических уравнений вида

$$F_i(X_1^{(j-1)}, X_2^{(j-1)}, \dots, X_n^{(j-1)}) + \sum_{k=1}^n \frac{\partial F_i}{\partial X_k} \Big|_{X_1^{(j-1)}, X_2^{(j-1)}, \dots, X_n^{(j-1)}} \cdot (X_k^{(j)} - X_k^{(j-1)}) = 0, \quad i = 1, \dots, n$$

относительно $\{X_k^{(j)}\}$.

Другое, возможно, более удобное представление той же системы выглядит так:

$$H(X^{(j-1)}) X^{(j)} = H(X^{(j-1)}) X^{(j-1)} - F(X^{(j-1)}),$$

где $H(X)$ — матрица частных производных системы как функция вектора X .

Как считать производные

Для получения матрицы частных производных можно воспользоваться простейшей формулой численного дифференцирования, но при этом возникает стандартный вопрос — как выбрать величину приращения аргумента. Обсудим это немного подробнее на примере функции одной переменной (для простоты).

Если есть некая функция $f(x)$, то при вычислении ее значений у нас получится (в худшем возможном случае) $\tilde{f}(x) = f(x)(1 + \varepsilon(x))$, где $\varepsilon(x)$ — машинная относительная погрешность представления результата (для простоты можно считать, что она по абсолютной величине константа и равна ε).

При численном дифференцировании получаем:

$$\tilde{f}'(x) = \frac{\tilde{f}(x + \Delta x) - \tilde{f}(x)}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} + f(x) \frac{\varepsilon(x + \Delta x) - \varepsilon(x)}{\Delta x}.$$

Тогда в самом худшем случае (если обе ошибки были «в разные стороны»), получится

$$\tilde{f}'(x) = \frac{\tilde{f}(x + \Delta x) - \tilde{f}(x)}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} + f(x) \frac{2\varepsilon}{\Delta x}.$$

Итого получается, что к абсолютной погрешности собственно линейной формулы дифференцирования, которая, естественно, равна $\frac{1}{2}f''(x)\Delta x$, добавляется еще и погрешность $f(x) \frac{2\varepsilon}{\Delta x}$, которая связана с машинной точностью. Если попытаться минимизировать функцию

$$E(\Delta x) = \frac{1}{2}f''(x)\Delta x + f(x) \frac{2\varepsilon}{\Delta x},$$

то получится, что минимум достигается при

$$\Delta x = 2\sqrt{\frac{\varepsilon f(x)}{f''(x)}}.$$

Если вычислительная задача имеет физический смысл и единицы измерения для аргумента и функции выбраны разумным образом, то $|f(x)| \sim |f''(x)| \sim 1$, а тогда оптимальной величиной приращения будет $\Delta x \sim \sqrt{\varepsilon}$. Как правило, подобный же выбор шага оказывается неплохим и в ситуации, когда про функцию мало что известно.

Представление данных и результатов

Требуется реализовать метод в виде функции или подпрограммы, а также тестирующую программу, использующую эту функцию.

Функция $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ задается в виде программной функции, принимающей в качестве аргументов вещественнозначный вектор X и возвращающей вещественнозначный вектор результата.

В функцию, реализующую метод Ньютона, передается название функции F (для Фортрана) или указатель на F (для C/C++). Кроме этого, функция, реализующая метод, должна получать как аргумент вектор начальных приближений и максимально возможное количество итераций, по достижении которого текущая итерация возвращается в качестве результата, даже если условие остановки итераций не выполнено.

Условие остановки итераций стандартное (достаточно малое изменение вектора на очередном шаге), метод решения СЛАУ выберите сами (нужно использовать уже написанные ранее функции/подпрограммы).

Тестирующая программа должна выводить на экран модуль вектора $F(X)$, где X — найденное решение. Вектор X нужно вывести в файл *result.dat* (по одному элементу в строке).