Санкт-Петербургский политехнический университет имени
Петра Великого

Институт прикладной математики и механики

Кафедра прикладной математики

Интервальный анализ
Отчёт по курсовой работе

Выполнил:

Студент: Дамаскинский Константин
Группа: 3630102/70201

Принял:

к. ф.-м. н., доцент
Баженов Александр Николаевич

2021 г.

Содержание

1. Постановка задачи

Исследователь поведение субдифференциального метода Ньютона при решении ИСЛАУ с матрицей:

$$\mathbf{A} = \begin{pmatrix} [3,4] & [5,6] \\ [-1,1] & [-3,1] \end{pmatrix} \tag{1.0.1}$$

и правой частью:

$$\mathbf{b} = \begin{pmatrix} [-3,4] \\ [-1,2] \end{pmatrix} \tag{1.0.2}$$

Объяснить, по каким причинам метод расходится при любом допустимом релаксационном параметре $\tau \in [0;1]$.

2. Теория

2.1. Субдифференциальный метод Ньютона

Кратко напомним суть субдифференциального метода Ньютона.Итерация выглядит следующим образом:

$$x^{(k)} \leftarrow x^{(k-1)} - \tau(D^{(k-1)})^{-1}\mathcal{F}(x^{(k-1)})$$

где $D^{(k-1)}\mathcal{F}(x^{(k-1)})$ – какой-нибудь субградиент в $x^{(k-1)}$, $\tau \in [0;1]$ – постоянный коэффициент. Алгоритм заканчивает работу, когда $\|\mathcal{F}(x^{(k)})\| < \varepsilon$.

В качестве такой начальной точки обычно принимают решение точечной СЛАУ:

$$(\text{mid}\mathbf{C})\tilde{x} = \mathtt{sti}(\mathbf{d}) \tag{2.1.1}$$

([2, стр. 607])

2.2. Вычисление субградиента

Для вычисления субградиента используются формулы:

$$\partial\mathcal{F}_i = -\sum_{j=1}^{n} \partial(\underline{\mathbf{c}_{ij}[-x_j, x_{j+n}]}),\ i = \overline{1,n} \tag{2.2.1}$$

$$\partial\mathcal{F}_i = \sum_{j=1}^{n} \partial(\overline{\mathbf{c}_{ij}[-x_j, x_{j+n}]}),\ i = \overline{n+1,2n} \tag{2.2.2}$$

Более детальное описание можно найти в [2] на стр. 610-613.

2.3. Приближённое решение ИСЛАУ с точечной прямоугольной матрицей

2.3.1. Первый вариант

В случае переопределённой матрицы самым простым способом решения задачи будет произвольный выбор строк матрицы ИСЛАУ и соответствующих компонент правого столбца с проверкой соблюдения условий сходимости (неособенности модуля выбранной матрицы).

Можно произвести этот процесс несколько раз. В таком случае задействуется большее количество уравнений из системы, так, чтобы по его окончании в решении оказались задействованы все условия. Тогда итоговое решение можно представить как пересечение всех найденных.

Недостатки данного подхода следующие:

1. Он не детерминирован. Однажды получив удачную конфигурацию матриц, можно более её не получить, если, конечно, не логгировать данные такого рода.

2. Эту конфигурацию в принципе получить тяжело: исключая из матрицы высоты 256 36 линейно независимых строк, можно очень долго произвольным образом выискивать следующую группу линейно независимы строк. И с огромной вероятностью в конце концов таких групп просто не останется.

Поэтому был применён упрощённый подход: на каждой итерации выбиралась произвольная абсолютно неособенная матрица и соответствующий правый столбец. При этом не вводилось требование задействовать все возможные строки.

2.3.2. Второй вариант

Другой подход – поэтапное исключение строк, не понижающих ранг рассматриваемой матрицы: на каждой итерации делается попытка удалить одну строку. Если удаление не понизило ранг, то оно делается окончательно, и процесс переходит на следующую строку. Если же удаление строки ведёт к понижению ранга матрицы, то эту строку мы не трогаем и движемся к следующей строке. Такой подход имеет следующие преимущества:

1. Детерминированность. Каждый запуск программы мы будем получать одно и то же решение.

2. Необходимо решать ИСЛАУ только один раз.

В итоге по причине детерминированности было принято решение использовать именно второй подход.

3. Реализация

Лабораторная работа выполнена в среде Octave в использованием портированного кода Сергея Петровича Шарого `subdiff` (см. “Приложения”). Операционная система Ubuntu 20.04.

Ссылка на исходный код лабораторной работы и отчёта находится в разделе “Приложения”.

4. Результаты

4.1. Задача 1

Субдифференциальный метод Ньютона вернул следующий результат:

$$\mathbf{x} = \begin{pmatrix} 1.000 \\ [1.444, 0.636] \\ [0.667, 1.273] \end{pmatrix} \tag{4.1.1}$$

Решение производилось с точностью до третьего знака. Невязка решения равна нулю (найденное решение является точным).

Замечание 1. Подразумевается ненулевая невязка при подстановке решения вплоть до последнего полученного знака. Однако предъявляется решение именно вплоть до третьего знака, поскольку запрос в методе был именно таковым. Однако, как мы знаем из теоретического раздела, данный метод отыскивает точное решение полиэдральных функций, поэтому полученный результат является отнюдь не удивительным, а закономерным.

4.2. Задача 2

Сопутствующие задачи с квадратными матрицами были решены с помощью метода, использующего знаково-блочные матрицы. В результате был получен следующий вектор:

$$\mathbf{x} = \begin{pmatrix} [0.620, 0.379] \\ [0.706, 0.783] \\ [1.045, 0.952] \\ [0.941, 1.067] \\ [0.972, 0.521] \\ [0.377, 0.624] \\ [0.430, 0.067] \\ [0.646, 0.354] \\ [0.818, 0.691] \\ [1.247, 0.253] \\ [0.462, 0.542] \\ [0.228, 0.272] \\ [0.287, -0.275] \\ [1.596, -1.096] \\ [0.737, 0.261] \\ [5.643, -4.642] \\ [0.693, -0.195] \\ [0.376, -0.388] \\ [-0.036, 0.038] \\ [0.558, -0.062] \\ [1.467, -0.467] \\ [1.927, -0.952] \\ [0.660, -0.160] \\ [0.242, -0.237] \\ [0.374, 0.126] \\ [0.516, 0.485] \\ [0.733, 0.766] \\ [0.781, 0.712] \\ [0.515, 0.485] \\ [0.494, 0.006] \\ [0.473, 0.526] \\ [1.056, 0.443] \\ [1.172, 0.827] \\ [0.928, 1.072] \\ [0.794, 0.704] \\ [0.388, 0.611] \end{pmatrix} \tag{4.2.1}$$

5. Обсуждение

В результате решения обеих задач были найдены формальные решения ИСЛАУ. Была экспериментально проверена сходимость субдифференциального метода Ньютона к точному решению в случае полиэдральной функции. Так-же была проверена путём подстановки корректность найденного формального решения второй задачи.

Список литературы

[1] А. Н. Баженов. *Лекции по интервальному анализу*. 2020.

[2] С. П. Шарый. *Конечномерный интервальный анализ*. Издательство “XYZ”, 2020. URL: http://www.nsc.ru/interval/Library/InteBooks/SharyBook.pdf.

6. Приложения

1. Репозиторий с кодом программы и кодом отчёта:

   https://github.com/kystyn/interval

2. `subdiff` Сергея Петровича Шарого для SciCodes:

   http://www.nsc.ru/interval/Programing/SciCodes/subdiff.sci