Дослідження Алгоритму Розв'язування Нелінійних Функціональних Рівнянь

Юрій Білущак
Відділ числових методів математичної фізики
Інститут прикладних проблем механіки і математики ім. Я.С. Підстригача НАНУ
Кафедра обчислювальної математики і програмування
Національний університет "Львівська політехніка",
Львів, Україна
byixx13@gmail.com

Ольга Чернуха
Відділ числових методів математичної фізики
Інститут прикладних проблем механіки і математики ім. Я.С. Підстригача НАНУ
Кафедра обчислювальної математики і програмування
Національний університет "Львівська політехніка",
Львів, Україна
zaliznuchna6@gmail.com

Анастасія Чучвара
Відділ числових методів математичної фізики
Інститут прикладних проблем механіки і математики ім. Я.С. Підстригача НАНУ
Львів, Україна
davydoka@gmail.com

Research of an Algorithm for Solving Nonlinear Functional Equations

Yurii Bilushchak
Department of numerical methods of mathematical physics
Pidstryhach Institute of Applied Problems of Mechanics and Mathematics of the National Academy of Sciences of Ukraine
Department of Computational Mathematics and Programming
Lviv Polytechnic National University
Lviv, Ukraine
byixx13@gmail.com

Olha Chernukha
Department of numerical methods of mathematical physics
Pidstryhach Institute of Applied Problems of Mechanics and Mathematics of the National Academy of Sciences of Ukraine
Department of Computational Mathematics and Programming
Lviv Polytechnic National University
Lviv, Ukraine
zaliznuchna6@gmail.com

Anastasiia Chuchvara
Department of numerical methods of mathematical physics
Pidstryhach Institute of Applied Problems of Mechanics and Mathematics of the National Academy of Sciences of Ukraine
Lviv, Ukraine
davydoka@gmail.com

***Анотація*— У роботі розглянуто алгоритм розв'язування нелінійних функціональних рівнянь на основі композиції методу простої ітерації та модифікації методу дихотомії. Показано використання алгоритму на прикладі знаходження розв'язків лінійної та нелінійної функцій.**

***Abstract*— In the paper the algorithm for solving nonlinear functional equations based on the composition of the simple iteration method and the modification of the dichotomy method is considered. The use of the algorithm is demonstrated through examples of finding solutions for both linear and nonlinear functions.**

***Ключові слова*—алгоритм; нелінійне функціональне рівняння; метод простої ітерації; метод дихотомії**

***Keywords*— algorithm; nonlinear functional equation; simple iteration method; dichotomy method**

I. Вступ

Для розв'язання нелінійних функціональних рівнянь у літературі представлені різні чисельні методи, наприклад, метод Дафтардар-Гейджі та Джафарі (DGJ) [1], метод декомпозиції Адоміана (ADM) [2], метод гомотопічного збурення (HPM) [3], тощо. Водночас, зазначені методи не позбавлені своїх обмежень та складнощів у їх застосуванні. Наприклад, метод гомотопічного збурення передбачає побудову гомотопій та накладання відповідних умов, що значно ускладнює обчислення; ADM метод передбачає обчислення многочленів Адоміана, що також потребує значних обчислювальних ресурсів [4].

У роботі запропоновано метод розв'язання нелінійних функціональних рівнянь на основі композиції методу простої ітерації та модифікації методу дихотомії [5] та досліджено його застосування для різного типу функцій.

II. Опис алгоритму

У багатьох прикладних задачах, наприклад, під час визначення довговічності роботи засипних фільтрів води, виникає необхідність чисельного розв'язання нелінійних функціональних рівнянь виду

$$\sup_{x\in\Omega} f\left(x,\tau_*\right)=M\ , \tag{1}$$

де $f(x,\tau)$ – відома функція двох змінних, τ_* – шукана величина, M – відома стала, $\Omega\in[0;\tau_*]$. Для розв'язання такого типу нелінійних функціональних рівнянь розвинуто метод на основі композиції методу простої ітерації та модифікації методу дихотомії. Його алгоритм є наступним [6]:

1. Задається початкове значення змінної τ_0, кроку розбиття h_l , точності обчислень ε та $k\in N$.
2. Перевіряється умова зупинки алгоритму

$$\left|M-f\left(x,\tau_i\right)\right|<\varepsilon\ .$$

У випадку, якщо вона справджується, то поточне значення τ_i є наближеним розв'язком необхідної точності і алгоритм завершується. У протилежному випадку здійснюється перехід на наступний крок.

3. Визначається знак виразу $M-f\left(x,\tau_i\right)$. Якщо ця величина додатна, то задаємо $\tau_i=\tau_{i-1}+h_l$. Протилежний випадок означає, що алгоритм вийшов за область значень функції $f\left(x,\tau_i\right)$ (оскільки $0\le f\left(x,\tau_i\right)\le M$), і повертаємось до попередньої точки τ_{i-1}. Після чого застосовуємо метод простої ітерації, зменшуючи крок розбиття $h_l=h_{l-1}/k$. Тоді $\tau_i=\tau_{i-2}+h_l$.

4. Обчислюється значення функції $f\left(x,\tau_i\right)$ у новій точці τ_i та повертаємось до кроку 2.

В загальному випадку k може бути різним для різних етапів знаходження кореня $k=k_j$. При цьому має виконуватись умова $k>1$. Оскільки, якщо $k=1$, то відбувається зациклення методу. Зауважимо, що оскільки $\tau_*=\tau_*(x)$, то звідси випливає, що край відрізка $[0;\,\tau_*(x)]$ $\tau=\tau_*$ може бути рухомим.

Даний алгоритм відноситься до класу лінійних. Складність даного алгоритму визначається сумарною кількість кроків ітераційного процесу $\Theta(M)$, де $M=m_0+m_1+m_2+\ldots+m_l$ – сумарна кількість кроків ітераційного процесу, m_i $(\,i=\overline{1,l}\,)$ – кількість кроків на рівномірному проміжку (етапі).

III. Застосування алгоритму для лінійної функції

Для перевірки ефективності роботи алгоритму застосуємо його до розв'язування достатньо простих рівнянь $f(\tau)=0$, коли $f(\tau)$ є лінійною і нелінійною функціями.

Нехай $f(\tau)$ є лінійною функцією $f(\tau)=10-\tau$, тобто потрібно знайти розв'язок рівняння $10-\tau=0$. Задавалися такі параметри алгоритму: $h=0.2$; $k=9$; $\varepsilon=0.001$; $\tau_0=0.01$.

В результаті отриманий розв'язок $\tau_*=9.99991007468$; програма використала $i=72$ етапів для досягнення заданої точності, при цьому 5 разів відбулось зменшення кроку h у 9 разів ($k=9$), останній крок $h_5=0.000030483158$.

На рис. 1 наведений графік функції $f(\tau)=10-\tau$, а на рис. 2 – результат роботи алгоритму у цьому випадку.

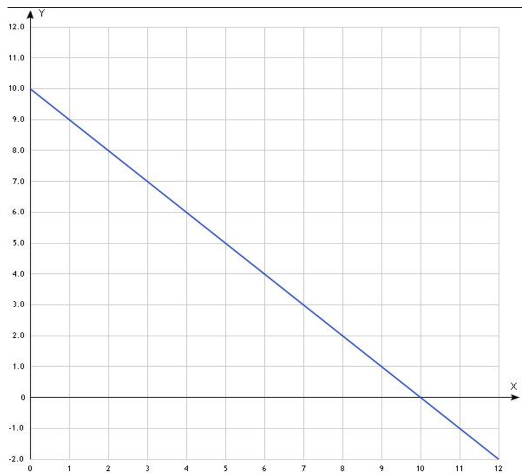

Рис. 1.Графік лінійної функції» $f(\tau)=10-\tau$

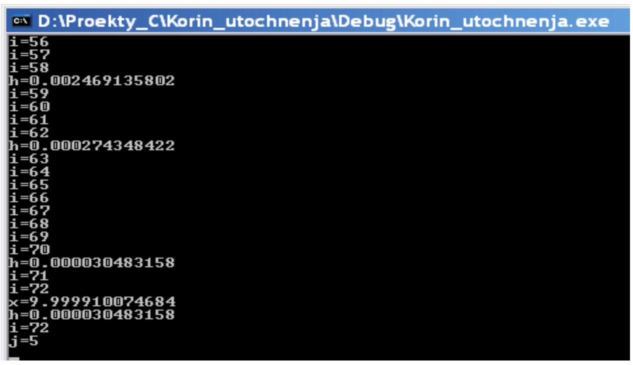

Рис. 2. Результат роботи алгоритму при розв'язуванні рівняння $10-\tau=0$

Розглянемо випадок, коли $f(\tau)$ є нелінійною функцією. А саме $f(\tau) = 9 - x^3 + x^2$. Тоді потрібно знайти розв'язок рівняння $9 - x^3 + x^2 = 0$. Задавалися такі параметри алгоритму: $h = 0.2$; $k = 9$; $\varepsilon = 0.00001$; $\tau_0 = 0.01$. Отримано такий розв'язок $\tau_* = 2.4723778871$; програма використала $i = 41$ етап для досягнення заданої точності, при цьому 7 разів відбулось зменшення кроку h у 9 разів ($k = 9$), останній крок $h_7 = 0.000000376335$.

На рис. 3 наведений графік функції $f(\tau) = 9 - x^3 + x^2$, а на рис. 4 – результат роботи алгоритму у цьому випадку.

Зазначимо, що кількість кроків такого типу алгоритмів суттєво залежить від початкового наближення (значення τ_0). При тестуванні даного алгоритму за «невдалого» вибору τ_0 – в околі нуля, для досягнення точності знадобилося 72 кроки та 5 переходів на метод простої ітерації для лінійної функції і 41 крок та 7 переходів – для нелінійної при $k = 9$.

Менша кількість кроків при суттєвому підвищенні точності пояснюється значно крутішим спаданням нелінійної функції (рис. 1 і 3), і відтак, швидшим наближенням до розв'язку τ_*.

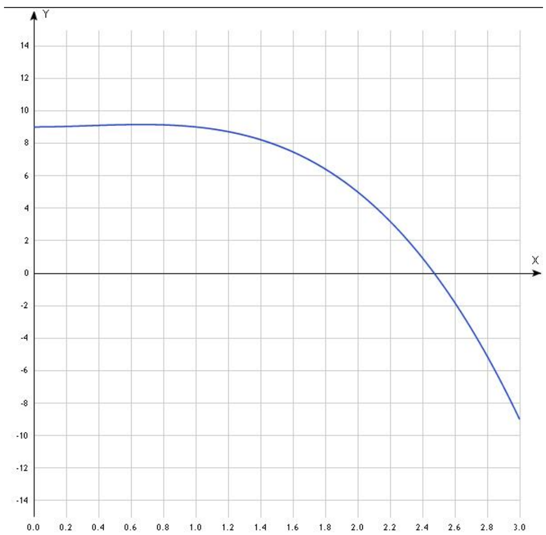

Рис. 3. Графік нелінійної функції» $f(\tau) = 9 - x^3 + x^2$

Зазначимо, що розвинений метод розроблявся для монотонно спадних функцій, оскільки саме така поведінка функцій $F \equiv M - \sup\limits_{x\in\Omega} f(x, \tau_*)$ є фізично обґрунтованою в переважній більшості випадків при встановлені часу насичення промислових фільтрів води.

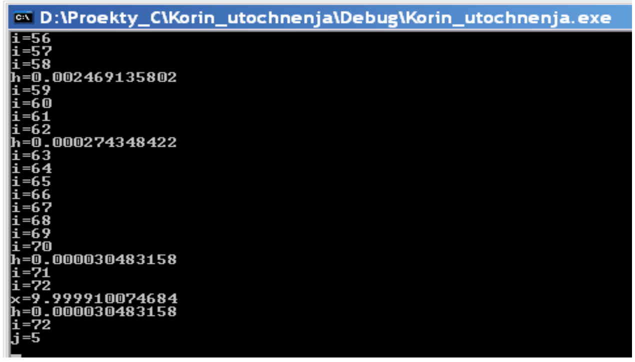

Рис. 4.Результат роботи алгоритму при розв'язуванні рівняння $9 - x^3 + x^2 = 0$

Проте можлива і інша поведінка функцій, зокрема F може дуже повільно виходити на стаціонарний режим або мати квазістаціонарні режими. Тоді в першому випадку, якщо цей стаціонарний режим не попадає в межі точності зупинки алгоритму, то потрібно накладати додаткову умову, щоб уникнути його зациклення, наприклад

$$\left| F(\tau_i) - F(\tau_{i+1}) \right| \le \varepsilon \qquad (2)$$

для всіх послідовних значень $i = k_1, ..., k_2$, де k_1 – перше значення i, коли виконується умова (2), значення k_2 задається. За виконання цієї умови алгоритм припиняє роботу та видає повідомлення, що розв'язку не досягнуто.

Якщо стаціонарне значення функції F задовольняє умову припинення роботи алгоритму ($\left| M - f(x, \tau_i) \right| < \varepsilon$), тобто ми отримуємо множину нулів функції F, то вважаємо розв'язком рівняння те значення аргументу, в якому вперше задовольняється умова (2). Тоді припиняється робота алгоритму. Проте для задач масоперенесення забруднення у фільтрах важливим є знаходження найменшого значення τ (часу), за якого відбувається насичення, коли через шар протікає забруднена речовина [7]. Тому в таких випадках пропонуємо робити додаткову перевірку досягнення точності через повернення назад на неповний крок (наприклад $0.1\, h_l^{(j)}$).

Можлива ще така поведінка функції F, коли вона досягає нуля в одній точці (або на дуже малому інтервалі), а потім відбувається її зростання – наприклад, осцилюючі функції. При цьому виникає небезпека пропустити перший нуль через вибір занадто великого кроку. Для вирішення цієї проблеми запропоновано два шляхи:

1. Вводимо вхідні дані, у тому числі задаємо крок, та запускаємо роботу алгоритму. Знаходимо нуль функції F із заданою точністю. Зменшуємо перший крок розбиття h_0 на порядок, і знову запускаємо роботу алгоритму при тих самих решта вхідних даних. Знаходимо нуль функції F із цією самою точністю при суттєво меншому кроці розбиття. Проводимо порівняння двох отриманих значень нулів

функції F. Якщо вони співпадають з певною точністю ε_1, то вважаємо, що нуль функції F або корінь рівняння $F = 0$ знайдено. Якщо ні, то знову зменшуємо перший крок розбиття h_0 на порядок, і запускаємо роботу алгоритму. Шукаємо корінь рівняння $F = 0$ і порівнюємо з попереднім значенням. Таку процедуру продовжуємо доки не досягнемо заданої точності ε_1.

2. На кожному кроці роботи алгоритму робимо додаткову перевірку, чи на пройденому інтервалі не відбувається зміна поведінки функції F. А саме: запускаємо роботу алгоритму, знаходимо значення F на першому кроці і встановлюємо, чи є ця точка нулем. Перевіряємо значення функції F цьому пройденому інтервалі, тобто цей інтервал розбиваємо на задану кількість підінтервалів і порівнюємо значення функції в цих точках. Якщо функція є монотонною, сталою або зростаючою, то переходимо на наступний крок алгоритму $\tau_i = \tau_{i-1} + h_l$. В іншому випадку вважаємо, що можлива наявність локального мінімуму, і відповідно можливо знаходження нуля функції на цьому інтервалі. Ділимо цей інтервал на більшу кількість під інтервалів і шукаємо чи досягає функція F нуля із заданою точністю. Якщо так, то завершуємо роботу алгоритму. Якщо ні, то переходимо на наступний крок алгоритму $\tau_i = \tau_{i-1} + h_l$.

IV. Висновки

Запропонований метод на основі композиції методу простої ітерації та модифікації методу дихотомії розроблявся для монотонно спадних функцій, оскільки різниці концентрацій насичення і функцій концентрацій сорбованої речовини у шаруватих фільтрах є саме монотонно спадними функціями. Проте за врахування наведених доповнень до алгоритму цей метод застосовний для знаходження нулів довільних неперервних функцій. Даний метод не вимагає значних обчислювальних потужностей, оскільки не потребує додаткових обчислень (обчислень гомотопій, чи многочленів Адоміана), проте є достатньо швидким та простим у реалізації. Запропонований алгоритм використано при розробці програмного комплексу «Wodfil», який призначено для аналізу ефективності роботи засипних фільтрів води.

Література References

[1] V. Daftardar-Gejji, H. Jafari, “An iterative method for solving nonlinear functional equations”, J. Math. Anal. Appl., vol. 316(2), 2006, pp.753–763.

[2] G. Adomian, Solving Frontier Problems of Physics: The Decomposition Method, Kluwer, Boston,1994.

[3] J.-H. He, “Homotopy perturbation technique”, Comput. Methods Appl. Mech. Eng., vol 178(3), 1999, pp. 257–262.

[4] K. Manoj, J. Aman, G. Varsha, “New Algorithm for Solving Non-linear Functional Equations”, Int. J. Appl. And Comput. Math., 2020, 6. 10.1007/s40819-020-0774-0.

[5] Ю. Білущак, В. Гончарук, О. Чернуха, А. Чучвара, “Числові методи для комп’ютерного моделювання довговічності роботи двошарового фільтра води”, in “Фізичні процеси та поля технічних та біологічних об’єктів”, Матеріали XV Міжнар. наук.-техн. конф., КрНУ ім. Михайла Остроградського, Кременчук, 2016, С. 118-120.

[6] О. Чернуха, Ю. Білущак, “Математичне та комп’ютерне моделювання процесів конвективної дифузії у двошарових засипних фільтрах води”, in “Інформаційні технології та комп’ютерне моделювання”, Івано-Франківськ, 2017, С. 349-354.

[7] O. Chernukha, A. Chuchvara, Y. Bilushchak., P. Pukach, N. Kryvinska, “Mathematical modelling of diffusion flows in two-phase stratified bodies with randomly disposed layers of stochastically set thickness”,