Laboratorium 11

Rozwiązywanie równań różniczkowych zwyczajnych

Adam Biśta, 28.05.2023

1 Treść zadań

Zadania

1. **Dane jest równanie różniczkowe (zagadnienie początkowe):**

$$y' + ycosx = sinxcosx$$

$$y(0) = 0$$

Znaleźć rozwiązanie metodą Rungego-Kutty i metodą Eulera. Porównać otrzymane rozwiązanie z rozwiązaniem dokładnym $y(x) = e^{-sinx} + sinx\breve{}1$.

2. **Dane jest zagadnienie brzegowe:**

$$y'' + y = x$$

$$y(0) = 1$$

$$y(0.5\pi) = 0.5\pi - 1$$

Znaleźć rozwiązanie metodą strzałów.
Porównać otrzymane rozwiązanie z rozwiązaniem dokładnym $y(x) = cos(x) - sin(x) + x$.

2 Rozwiązania zadań

1. • **Metoda Rungego-Kutty**

   Znana również **jako metoda Rungego-Kutty czwartego rzędu**, jest to renomowana metoda numeryczna stosowana do rozwiązywania równań różniczkowych zwyczajnych (ODE). Ta technika opiera się na sekwencyjnym procesie, który przybliża rozwiązanie danego równania różniczkowego poprzez etapową aktualizację wartości funkcji w kolejnych punktach.

   Algorytm Rungego-Kutty polega na wyliczaniu średnich wartości nachylenia funkcji w kilku punktach w ramach danego kroku czasowego. Najczęściej stosowaną formą jest metoda czwartego rzędu, która polega na wyliczeniu czterech takich nachyleń.

   Niech **h** to długość kroku całkowania.
   Iteracyjny wzór na **y** zgodnie z metodą Rungego-Kutty czwartego rzędu prezentuje się następująco:

   $$y_{n+1} = y_n + \Delta y$$
   $$\Delta y_n = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

   Wiemy, że:

   $$k_1 = h \cdot f(x_n, y_n),$$
   $$k_2 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right),$$
   $$k_3 = h \cdot f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right),$$
   $$k_4 = h \cdot f(x_n + h, y_n + k_3).$$

   Jest widoczne, że wartość y_{n+1} jest zależna od wartości y_n i h.
   Podobnie jak w przypadku innych technik iteracyjnych stosowanych do rozwiązywania równań różniczkowych, metoda Rungego-Kutty pozwala generować serię punktów, które stopniowo zbliżają się do docelowego rozwiązania.

   • **Metoda Eulera**

   Metoda Eulera jest jednym z najbardziej podstawowych narzędzi numerycznych do rozwiązywania zwyczajnych równań różniczkowych (ODE). Działa na zasadzie estymowania wartości funkcji w kolejnych momentach czasowych, bazując na jej pochodnej.

Metoda Eulera stanowi specjalny przypadek metod Rungego-Kutty. Tradycyjnie, pomimo że powstała wcześniej, zachowuje tę nazwę. Głównym elementem odróżniającym te metody jest iteracyjna zależność:

$$y_{n+1} = y_n + k_1$$

Poniższy program w pythonie rozwiązuje zadanie obydwoma metodami:

```
from math import sin, cos, e

def f(x, y):
    return sin(x) * cos(x) - y * cos(x)

def exact_solution(x):
    return e ** (-sin(x)) + sin(x) - 1

def runge_kutta(iterations, h, x_0, y_0):
    x = x_0
    y = y_0
    for _ in range(iterations):
        k1 = h * f(x, y)
        k2 = h * f(x + h / 2, y + k1 / 2)
        k3 = h * f(x + h / 2, y + k2 / 2)
        k4 = h * f(x + h, y + k3)
        x += h
        y += (k1 + 2 * k2 + 2 * k3 + k4) / 6
    return x, y

def euler(iterations, h, x_0, y_0):
    x = x_0
    y = y_0
    for _ in range(iterations):
        k1 = h * f(x, y)
        x += h
        y += k1
    return x, y

def main():
    print("Metoda Rungego-Kutty, błąd bezwzględny")
    for n in [100, 1000, 10000, 100000, 1000000]:
        x, y = runge_kutta(n, 1 / n, 0, 0)
        print(f"{abs(y - exact_solution(x))}")
```

```
    print("Metoda Eulera, błąd bezwzględny")
    for n in [100, 1000, 10000, 100000, 1000000]:
        x, y = euler(n, 1 / n, 0, 0)
        print(f"{abs(y - exact_solution(x)):.10f}")

if __name__ == "__main__":
    main()
```

Wyniki wywołania:

Metoda	Ilość iteracji	Błąd bezwzględny
Rungego-Kutty	100	1.9702350861905416e-11
Rungego-Kutty	1000	1.9984014443252818e-15
Rungego-Kutty	10000	2.248201624865942e-14
Rungego-Kutty	100000	6.196709811945311e-13
Rungego-Kutty	1000000	3.0789815141929466e-12
Eulera	100	0.0007407326
Eulera	1000	0.0000733964
Eulera	10000	0.0000073329
Eulera	100000	0.0000007332
Eulera	1000000	0.0000000733

Wnioski

Metoda Eulera, będąca fundamentalnym narzędziem w dziedzinie równań różniczkowych, cechuje się prostotą zarówno w zrozumieniu, jak i wdrożeniu. Ta cecha czyni ją często stosowanym narzędziem do przedstawiania podstawowych koncepcji numerycznego rozwiązywania równań różniczkowych. Niemniej jednak, mimo jej prostoty, metoda Eulera ma swoje ograniczenia, w tym ograniczoną dokładność, zwłaszcza w przypadku równań o skomplikowanej dynamice. Ponadto, metoda Eulera może prowadzić do nagromadzenia błędów, co może skutkować nierealistycznymi wynikami dla pewnych typów równań.

Z drugiej strony, metoda Rungego-Kutty zdobyła szerokie uznanie za sprawą swojej dokładności i skuteczności w rozwiązywaniu zróżnicowanych rodzajów równań różniczkowych. Metoda ta jest nie tylko efektywna, ale również elastyczna, co pozwala na dostosowanie jej do konkretnej sytuacji. Dzięki temu, metoda Rungego-Kutty jest zdecydowanie niejednorodna - w zależności od konkretnych wymagań, można zastosować różne jej odmiany, w tym metoda Rungego-Kutty drugiego rzędu. Kluczową zaletą tej metody jest jej zdolność do adaptacji do różnorodnych problemów, co sprawia, że jest to niezmiernie uniwersalne narzędzie w dziedzinie równań różniczkowych.

2. Metoda strzałów

Technika numeryczna o nazwie metoda strzałów, czasem nazywana również metodą strzałów początkowych, jest używana do rozwiązywania równań różniczkowych zwyczajnych (ODE), a w szczególności w kontekście problemów brzegowych.
Kluczowym elementem tej metody jest dążenie do ustalenia odpowiednich warunków początkowych, które pozwolą na spełnienie warunków brzegowych w kontekście określonego równania różniczkowego.
Zasadniczo metoda strzałów odwraca problem, przekształcając zagadnienie brzegowe w zagadnienie początkowe w celu jego rozwiązania.

Dla problemu brzegowego drugiego rzędu równania różniczkowego, zapisujemy je jako:

$$y''(t) = f(t, y(t), y'(t)), \quad y(t_0) = y_0, \quad y(t_1) = y_1 \tag{1}$$

Rozważmy teraz $y_a(t)$ jako rozwiązanie dla problemu początkowego. Zatem, mamy:

$$y''(t) = f(t, y(t), y'(t)), \quad y(t_0) = y_0, \quad y'(t_0) = a \tag{2}$$

Możemy zdefiniować funkcję $F(a)$ jako różnicę pomiędzy $y_a(t)$ a ustaloną wartością brzegową y_1, co daje:

$$F(a) = y_a(t_1) - y_1 \tag{3}$$

W przypadku, gdy rozwiązanie problemu brzegowego istnieje, funkcja F posiada pierwiastek, który odpowiada wartości $y'(t_0)$ prowadzącej do rozwiązania $y(t)$ problemu brzegowego. Aby odnaleźć wartość a, stosuje się standardowe metody numeryczne, takie jak metoda bisekcji czy metoda Newtona.

Cały proces metody strzałów można porównać do strzelania w cel - korzystamy z początkowych wartości jako "strzału", następnie je modyfikujemy, aby spełnić określone warunki brzegowe.

Metoda strzałów jest szczególnie użyteczna dla problemów brzegowych w równaniach różniczkowych, gdy nie mamy dostępu do analitycznych rozwiązań. Z właściwie dobranymi wartościami początkowymi i odpowiednią techniką numeryczną, umożliwia uzyskanie przybliżonego rozwiązania, które spełnia określone warunki brzegowe.

Program rozwiązujący zadanie metodą strzałów napisany w języku Python

```python
from math import sin, cos, pi

def f(x, y, y_prim):
    return x - y

def exact_solution(x):
    return cos(x) - sin(x) + x

def runge_kutta_hit(iterations, x_0, y_0, a, h, func=f):
    x = x_0
    y = y_0
    for _ in range(iterations):
        k1 = h * func(x, y, a)
        k2 = h * func(x + h / 2, y + k1 / 2, a)
        k3 = h * func(x + h / 2, y + k2 / 2, a)
        k4 = h * func(x + h, y + k3, a)
        delta_a = (k1 + 2 * k2 + 2 * k3 + k4) / 6
        k1 = h * a
        k2 = h * (a + h * func(x + h / 2, y + k1 / 2, a))
        k3 = h * (a + h * func(x + h / 2, y + k2 / 2, a))
        k4 = h * (a + h * func(x + h, y + k3, a))
        delta_y = (k1 + 2 * k2 + 2 * k3 + k4) / 6
        x += h
        y += delta_y
        a += delta_a
    return x, y

def hit_method(final_iterations, x_0, x_1, y_0, y_1, a_0,
    a_1, h, epsilon):
    bisect_iterations = int((x_1 - x_0) / h)
    a = (a_0 + a_1) / 2
    y = runge_kutta_hit(bisect_iterations, x_0, y_0, a,
        h)[1]
    i = 0
    while abs(y - y_1) > epsilon:
        if (y - y_1) * (runge_kutta_hit(bisect_iterations,
            x_0, y_0, a_0, h)[1] - y_1) > 0:
            a_0 = a
        else:
            a_1 = a
        a = (a_0 + a_1) / 2
        y = runge_kutta_hit(bisect_iterations, x_0, y_0, a,
            h)[1]
        i += 1
```

```
    return runge_kutta_hit(final_iterations, x_0, y_0, a, h)

def main():
    for n in [100, 1000, 10000, 100000, 1000000]:
        h = 1 / n
        x, y = hit_method(n, 0, pi / 2, 1, pi / 2 - 1, -100,
        ↪  100, h, 1e-3)
        print(f"n={n}, error={abs(y - exact_solution(x))}")

if __name__ == "__main__":
    main()
```

Wyniki wywołania programu:

Tabela 1: Błąd bezwzględny dla różnej liczby iteracji dla metody strzałów

n	**Błąd bezwzględny**
100	$8.697235448218432 \times 10^{-5}$
1000	$1.057578239682666 \times 10^{-5}$
10000	$1.076485546702699 \times 10^{-6}$
100000	$1.0783849135886925 \times 10^{-7}$
1000000	$1.0783245851797574 \times 10^{-8}$

Wnioski:

Metoda strzałów jest popularną techniką numeryczną do rozwiązywania równań różniczkowych, zwłaszcza równań różniczkowych zwyczajnych (ODE). Polega na przekształceniu zadania rozwiązania równania różniczkowego w zadanie rozwiązania równania algebraicznego. Metoda ta opiera się na iteracyjnym próbowaniu różnych warunków początkowych, aż zostanie znalezione przybliżone rozwiązanie spełniające zadane warunki brzegowe lub warunki początkowe.

Metoda strzałów jest elastyczna i potrafi radzić sobie z różnymi rodzajami równań różniczkowych, w tym z równaniami nieliniowymi i układami równań. Jednak może być czasochłonna, szczególnie dla skomplikowanych równań, dlatego ważne jest optymalne dobrane parametry i liczba iteracji.

Zastosowanie metody strzałów nie ogranicza się tylko do prostych równań różniczkowych, ale może być również stosowane do bardziej złożonych przypadków, włączając w to równania różniczkowe wyższego rzędu, które mogą być przekształcone na układy równań różniczkowych zwyczajnych. Metoda ta jest szeroko wykorzystywana w dziedzinach takich jak mechanika płynów, fizyka, inżynieria i nauki przyrodnicze, gdzie równania różniczkowe są powszechnie spotykane.

Podsumowując, metoda strzałów jest wartościowym narzędziem do rozwiązywania różnorodnych równań różniczkowych, ale wymaga starannego doboru parametrów i może być czasochłonna dla bardziej skomplikowanych przypadków.

3 Bibliografia

- Katarzyna Rycerz: "Wykład z przedmiotu Metody Obliczeniowe w Nauce i Technice"
- https://home.agh.edu.pl/ funika/mownit/lab11/
- https://pl.wikipedia.org/wiki/Algorytm-Rungego-Kutty
- https://pl.wikipedia.org/wiki/Metoda-Eulera
- https://pl.wikipedia.org/wiki/Metoda-strzalow