

Zadanie numeryczne 9

Oleg Semenov

Spis treści

Wstęp	2
Metoda interpolacji odwrotnej	2
Kod interpolacji odwrotnej	2
Metoda siecznych	2
Kod metody siecznych	3
Rozwiązanie zadania numerycznego	3
Klasa punktów	3
Funkcja	3
Funkcja Main i Wyniki	3
Przykładowy wynik	4

Wstęp

W rozwiązaniu danego zadania skorzystałem z Pythona 3.7 z dodatkowym użyciem biblioteki NumPy dla niektórych funkcji matematycznych jak $\cosh()$ oraz $\sinh()$. Tutaj wykorzystuję metody siecznych oraz interpolacji odwrotnej by znaleźć miejsce zerowe funkcji:

$$f(x) = (x^2 - 1) \sinh^3 x = 0.$$

Metoda interpolacji odwrotnej

Metoda interpolacji odwrotnej polega na stabelizowaniu wartości funkcji w węzłach (w danym przypadku w 3). Przy czym wartości mają być monotoniczne, czyli: $f_1 > f_2 > f_3$ lub $f_1 < f_2 < f_3$. Skoro funkcja jest monotoniczna, jest odwracalna (węzły i wartości zamieniają się miejscami). Na podstawie odwróconej tabeli tworzymy wielomian interpolacyjny i obliczamy jego wartość, czyli przybliżenia funkcji odwrotnej, w zerze.

Kod interpolacji odwrotnej

```
def inverse_interp(points):
    points = sorted(points)
    for point in points:
        point.x, point.y = point.y, point.x

    temp = 0

    while True:
        temp = 0
        for i, p1 in enumerate(points):
            inter = 1

            for j, p2 in enumerate(points):
                if i != j:
                    inter *= (-p2.x) / (p1.x - p2.x)

            temp += inter * p1.y

        points = points[1:3] + [Point(func(temp), temp)]

        if abs(func(points[2].y)) <= epsilon:
            break

    return temp
```

Metoda siecznych

Metoda siecznych to algorytm interpolacji liniowej. Polega na przyjęciu, że funkcja ciągła na dostatecznie małym odcinku w przybliżeniu zmienia się w sposób liniowy. Możemy wtedy na odcinku $\langle a, b \rangle$ krzywą $y = f(x)$ zastąpić sieczną.

Metodę siecznych dla funkcji $f(x)$, mającej pierwiastek w przedziale $\langle a, b \rangle$ można zapisać następującym wzorem rekurencyjnym:

$$\begin{cases} x_0 = a \\ x_1 = b \\ x_{n+1} = \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})} \end{cases}$$

Kod metody siecznych

```
def secant(p0, p1, N=1000, e=epsilon):
    x0, x1, x2, step = p0.x, p1.x, 0, 1
    condition = True
    while condition:
        if func(x0) == func(x1):
            break

        x2 = x0 - (x1 - x0) * func(x0) / (func(x1) - func(x0))
        x0 = x1
        x1 = x2
        step = step + 1

        if step > N:
            return "Non convergent"

        condition = abs(func(x2)) > e
    return x2
```

Rozwiązanie zadania numerycznego

Klasa punktów

Dla ułatwienia zaimplementowałem prostą klasę punktów:

```
class Point:
    def __init__(self, x, y):
        self.x, self.y = x, y

    def __lt__(self, other):
        return self.y < other.y
```

Funkcja

```
def func(x):
    return (pow(x, 2) - 1) * pow(np.sinh(x), 3)
```

Funkcja Main i Wyniki

```
if __name__ == '__main__':
    for k in range(10):
        points = []
        for i in range(3):
            random.seed()
            x = random.uniform(0, 1)
            y = func(x)
            points.append(Point(x, y))
        for i in range(3):
            print(f"P{i + 1}: ({points[i].x}, {points[i].y})")

        print(f"Metoda siecznych: {secant(points[0], points[2])}, "
              f"{func(secant(points[0], points[2]))}")
        print(f"Interpolacja odwrotna: {inverse_interp(points)}, "
              f"{func(inverse_interp(points))}\n")
```

Z wyników można wnioskować że istnieją dwa miejsca zerowe na przedziale $\langle 0, 1 \rangle$: 0 i 1.

Przykładowy wynik

P1: (0.20362145257747266, -0.008261729064385253)
P2: (0.3418810420190983, -0.037404690167779975)
P3: (0.888335590456658, -0.21716531420696691)
Metoda siecznych: 0.003848248984679096, -5.698837515475983e-08
Interpolacja odwrotna: 0.0042263227740731105, -6.448720502355558e-08

P1: (0.2922591203878193, -0.02382447279027775)
P2: (0.6195116243594119, -0.1770813671596562)
P3: (0.3419307394350022, -0.037420195413040355)
Metoda siecznych: 0.0041516151187552764, -7.155623994765022e-08
Interpolacja odwrotna: 0.003773606878639618, 3.994431839204296e-08

P1: (0.8564183934118679, -0.23953991441921554)
P2: (0.12054069206003237, -0.0017385935301212397)
P3: (0.8563496385811277, -0.23957456868710467)
Metoda siecznych: 1.0000000005654409, 1.8354978983514176e-08
Interpolacja odwrotna: 0.003830842209447111, -5.0616551074662515e-08

P1: (0.3024353081129474, -0.026305087533405498)
P2: (0.013286650768093544, -2.3453530055258488e-06)
P3: (0.4114888424539712, -0.06296062916664878)
Metoda siecznych: 0.004594663492269857, -9.699660746589883e-08
Interpolacja odwrotna: 0.004562836066774304, 2.037071795936142e-09

P1: (0.524375321168391, -0.11979921632410094)
P2: (0.23611911747299885, -0.012780959052482358)
P3: (0.8069380469150944, -0.2521202691626836)
Metoda siecznych: 0.004184684143884735, -7.327979531533871e-08
Interpolacja odwrotna: 0.0036813003575881737, -4.5049573922795706e-08

P1: (0.6576930502129487, -0.19980232105190004)
P2: (0.6996935473129631, -0.22247697045654033)
P3: (0.8626804740356128, -0.23615429378483518)
Metoda siecznych: -0.004102479622104678, 6.904554194236184e-08
Interpolacja odwrotna: 0.004518956544833934, 1.5027686351442974e-11

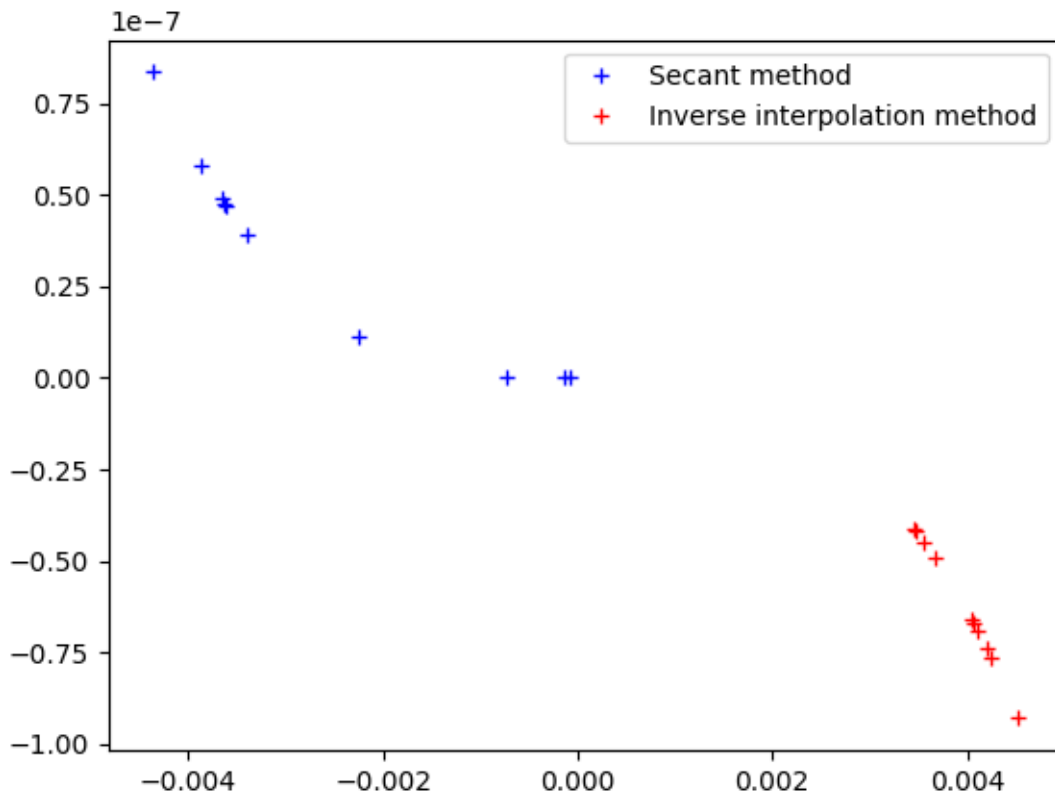
P1: (0.29978925965431535, -0.0256452886361712)
P2: (0.4336060369551925, -0.07267889335816777)
P3: (0.9402672746582279, -0.14806367641896973)
Metoda siecznych: 0.004635278426873101, -9.959162354178341e-08
Interpolacja odwrotna: 0.003459316540250148, -6.960630148663905e-08

P1: (0.974887376583946, -0.07286282172681363)
P2: (0.47665668158210395, -0.09368091646906196)
P3: (0.5671725077437076, -0.145109554854362)
Metoda siecznych: 0.003693551431857224, -5.038827490840608e-08
Interpolacja odwrotna: 0.999999999998577, -4.557061984125671e-08

P1: (0.41075117682268725, -0.06264956561636906)
P2: (0.3010123376310069, -0.02594896850962928)
P3: (0.9530100306898299, -0.12347330091476127)
Metoda siecznych: -0.004003172800385327, 6.415190121011366e-08
Interpolacja odwrotna: 0.0036263029912540178, -6.303645517609138e-08

P1: (0.9889241251274559, -0.034223675185991675)
P2: (0.8780803867178955, -0.22579418992883624)
P3: (0.4779612927629818, -0.09435788974757232)
Metoda siecznych: 0.00462895622906072, -9.918467383577005e-08
Interpolacja odwrotna: 0.9999999999982273, 3.850425913739391e-08

Inny przykład za pomocą wykresu:



za pomocą biblioteki matplotlib:

```
for k in range(10):
    points = []
    for i in range(3):
        random.seed()
        x = random.uniform(0, 1)
        y = func(x)
        points.append(Point(x, y))
    for i in range(3):
        print(f"P{i + 1}: ({points[i].x}, {points[i].y})")

    print(f"Metoda siecznych: {secant(points[0], points[2])}, "
          f"{func(secant(points[0], points[2]))}")
    print(f"Interpolacja odwrotna: {inverse_interp(points)}, "
          f"{func(inverse_interp(points))}\n")
    results_inv.append(inverse_interp(points))
    results_sec.append(secant(points[0], points[2]))

plt.plot(results_sec, [func(num) for num in results_sec], "b+",
         label="Secant method")
plt.plot(results_inv, [func(num) for num in results_inv], "r+",
         label="Inverse interpolation method")
plt.legend()
plt.show()
```