Aproksymacja równań różniczkowych - projekt

Natalia Wojtania i Grzegorz Chojnacki

8 stycznia 2021

1 Zadanie

1.1 Tytuł

Tytuł zadania to „Metody numeryczne dla zagadnieńróżniczkowych”.

1.2 Treść

Napisz program, który rozwiąże trzema metodami(Eulera, zmodyfikowaną Eulera oraz Heuna)zagadnienie różniczkowe:

$$y'(x) = f(x, y(x)), y(1) = 5,$$

gdzie

$$f(x, y(x)) = -5x^4 + 2\sqrt{y + x^5 - 5} + 5.$$

Program ma również obliczyć dokładność dla każdej z tych metod, porównując je z dokładnym rozwiązaniem:

$$y(x) = -x^5 + x^2 + 5.$$

1.3 Metody

W programie należy wykorzystać metodę Eulera, zmodyfikowaną metodę Eulera oraz metodę Heuna (udoskonaloną metodę Eulera).

1.3.1 Opis metod

!!!!! Metoda siecznych (interpolacji liniowej) polega na przyjęciu, że funkcja ciągła na dostatecznie małym odcinku w przybliżeniu zmienia się w sposób liniowy. Można wtedy na odcinku $[a, b]$ krzywą $y = f(x)$ zastąpić sieczną. Za przybliżoną wartość pierwiastka przyjmuje się punkt przecięcia siecznej z osią odciętych OX. Miejsce przecięcia tej prostej z osią X jest przybliżonym wynikiem szukanego miejsca zerowego, o ile różnica bezwzględna wartości z dwóch ostatnich iteracji jest mniejsza od założonej dokładności. Metoda ta wymaga ustalenia na przedziale $[a, b]$ dwóch punktów startowych x_0 i x_1. Metodę siecznych dla funkcji $f(x)$, mającej pierwiastek w przedziale $[a, b]$ można zapisać następującym wzorem rekurencyjnym:

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}, k \geqslant 1$$

i

$$x_0 = a, x_1 = b,$$

gdzie w każdym kroku x_{k+1} to miejsce zerowe siecznej wykresu $y = f(x)$ w punktach $(x_{k-1}, f(x_{k-1}))$ oraz$(x_k, f(x_k))$, czyli prostej

$$y = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}(x - x_k) + f(x_k)$$

1.3.2 Przykład

Dla zagadnienia różniczkowego $y'(x) = f(x, y(x)), y(1) = 5$, na przedziale [1,2] przyjmując $N = 2$ policzyć rozwiązanie trzema metodami oraz obliczyć dokładność dla każdej z metod, wiedząc, że dokładne rozwiązanie to
$y(x) = -x^5 + x^2 + 5$ oraz $f(x, y(x)) = -5x^4 + 2\sqrt{y + x^5 - 5} + 5$.

2 Opis implementacji algorytmu

Implementacja realizująca metody Eulera, zmodyfikowaną Eulara oraz Heuna.

2.1 Dane wejściowe

Na wejściu program pobiera od użytkownika liczbę n, określająca podział odcinka $[a, b]$ oraz liczbę b, informująca o końcu odcinka $[a, b]$.

2.2 Przebieg działania

!!! Program wyświetla komunikat: „Wprowadź dokładność rozwiązania $\epsilon \in (0, 1)$ ". Jeśli została wprowadzona prawidłowa wartość dokładności, to program poprzez funkcję *calculate* wylicza przybliżone rozwiązanie i dzięki funkcji *refresh* wyświetla je wraz z liczbą kroków. Próba wprowadzenia nieprawidłowych danych, które weryfikowane są w programie w funkcji *refresh* skutkuje wyświetleniem stosownego ostrzeżenia.

Następnie funkcja *calculate* klasy *SecantMethod* zajmuje się wyliczeniem przybliżonego rozwiązania w oparciu o podaną dokładność i przedział wyszukany za pomocą funkcji *findInterval*. Szukanie przedziału zaczyna się od $[0, 1]$ i jeżeli funkcja nie przechodzi w nim przez oś OX, to przedział jest poszerzany.
Funkcja *getNext*, której argumentami są a i b odpowiednio oznaczające x_{k-1} oraz x_k zwraca wartość poszczególnego x_{k+1}.
Funkcja *isGoodEnough* sprawdza czy różnica $|x_k - x_{k-1}|$ jest mniejsza od podanej przez użytkownika dokładności. Jeśli tak, to kończymy przekazując wynik oraz ilość kroków. W przeciwnym wypadku liczone jest kolejne przybliżenie tak długo, aż warunek zostanie spełniony.

Wynikiem działania programu jest przybliżone rozwiązanie równania: x_k oraz liczba wykonanych kroków: k.

2.3 Najważniejsze fragmenty programu

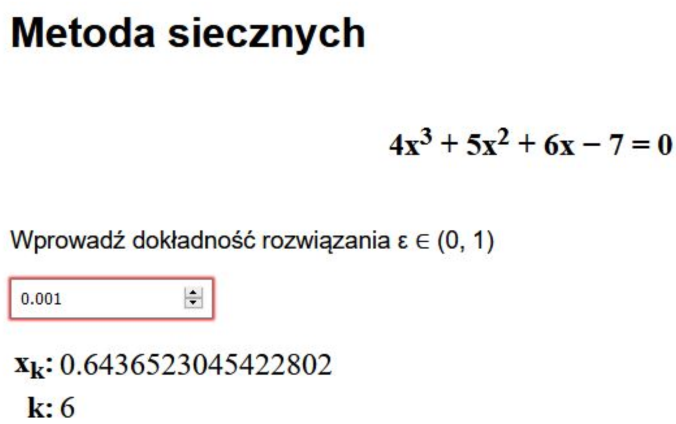
!!! secantMethod.js

```
class SecantMethod {
  f = x => 4*x**3 + 5*x**2 + 6*x - 7

  constructor(precision) {
    this.precision = precision
    this.interval = this.findInterval()
  }

  findInterval = (a = 0, b = 1) => this.f(a) * this.f(b) < 0
    ? [a, b]
    : this.findInterval(a - (b - a), b + (b - a))

  // a = x_{k-1}, b = x_{k}
  getNext = (a, b) => b - (this.f(b) * (b - a)) / (this.f(b) - this.f(a))
  isGoodEnough = (next, prev) => Math.abs(next - prev) < this.precision

  calculate() {
    const g = (a, b, steps = 2) => {
      const next = this.getNext(a, b)
      return this.isGoodEnough(next, b)
        ? ({ result: next, steps })
        : g(b, next, steps + 1)
    }
    return g(this.interval[0], this.interval[1])
  }
}
```

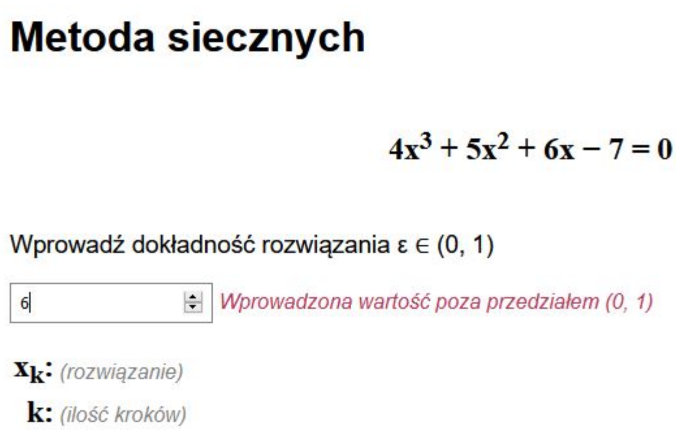
gui.js

```
const gui = new (class {
  input  = document.getElementById('input')
  result = document.getElementById('result')
  steps  = document.getElementById('steps')
  error  = document.getElementById('error')

  refresh() {
    const precision = this.getPrecision()
    if (0 < precision && precision < 1) {
      this.clearError()
      const answer = new SecantMethod(precision).calculate()
      this.result.innerText = answer.result
      this.steps.innerText  = answer.steps
    } else this.setError()
  }

  setError()   { this.error.innerText  = 'Wprowadzona wartość poza przedziałem (0,
  clearError() { this.error.innerText  = '' }

  update = debounce(() => this.refresh(), 10)

  getPrecision = () => Number.parseFloat(this.input.value)

})()
```

2.4 Widok działania programu



Rysunek 1: Prawidłowo wprowadzone dane



Rysunek 2: Nieprawidłowo wprowadzone dane