Отчет по лабораторной работе №5

Снопов П.М.

15 мая 2020 г.

3 курс, 3 группа

Лабораторная работа №5
Метод универсальной дифференциальной прогонки для линейных уравнений второго порядка.
Вариант 9

1. Постановка задачи Метод универсальной дифференциальной прогонки для линейных уравнений второго порядка:

Имеется линейное дифференциальное уравнение второго порядка $y''(x)+p(x)y'(x) = q(x)y(x) + f(x)$, где $x \in [A, B]$ с краевыми условиями $y(C) = y_c$, где С $\in \{A,B\}$. Необходимо численно решить ОДУ с заданной точностью и автоматическим выбором длины шага.

2. Метод решения Решать задачу будем методом Рунге-Кутта второго порядка, для уточнения решения будем пользоваться методом Рунге-Кутта четвертого порядка. Явный метод Рунге-Кутта задается формулами:

$$\begin{aligned}
&y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i \\
&k_1 = f(x_n, y_n) \\
&k_2 = f(x_n + c_2 h, y_n + a_{21} h k_1) \\
&... \\
&k_s = f(x_n + c_s h, y_n + \sum_{p=1}^{s-1} a_{sp} h k_p)
\end{aligned} .$$

Где h – величина шага сетки по x.

Для решения данной задачи воспользуемся следующим методом второго порядка:

$$\begin{aligned}
&y_{n+1} = y_n + k_2 \\
&k_1 = hf(x_n,y_n) \\
&k2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)
\end{aligned}$$

А для уточнения данной задачи воспользуемся следующим методом четвертого порядка:

$$\begin{aligned}
&y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
&k_1 = hf(x_n,y_n) \\
&k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\
&k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \\
&k_4 = hf(x_n + h, y_n + k_3)
\end{aligned}$$

Для начала пусть $h := \frac{B-A}{10}$. Далее на каждом шагу определяем рекомендуемую длину шага следующим образом:

$$h_e := \sqrt[\frac{1}{4}]{\frac{\varepsilon_n}{\varepsilon}}h$$

где $\varepsilon_n := |\hat{y}_n - \tilde{y}_n|$, где $\hat{y}_n$ – значение, полученное методом второго порядка, а $\tilde{y}_n$– методом четвертого порядка. Далее проводим выбор шага следующим образом:

$$h_n = \begin{cases} h_{min}, & \text{если } h_e < h_{min} \\ h_e, & \text{если } h_{min} < h_e < h_{max} \\ h_{max}, & \text{если } h_e > h_{max} \end{cases}$$

3. Основные процедуры Основные функции, используемые при решении задачи:

```
def estimate(f: Callable, h: float, x: float, y=0) -> float:
```

Функция, соответствующая методу Рунге-Кутта второго порядка

```
def optimize(f: Callable, h: float, x: float, y=0) -> float:
```

Функция, соответствующая методу Рунге-Кутта четвертого порядка

```
def get_data(name: str) -> list:
```

Функция для получения данных

```
def give_data(X:list, Y:list):
```

Функция записывающая полученные данные

```
def solve(f: Callable) -> list:
```

Основная функция, осуществляющая решение задачи.

3. Графики полученного численного решения и точного