Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
факультет программной инженерии и компьютерной техники

Лабораторная работа №5

по дисциплине

Вычислительная математика

Выполнил:
Студент группы P3230
Толстых М.А.

Преподаватель:
Перл И. А.
Перл О. В.

Санкт-Петербург, 2024

Описание метода

Метод Адамса.

Задается отрезок [a; b], функция и точность ε, y(a).

$$\frac{y-y_i}{h} = \sum_{j=0}^{k} \beta_i f_{i+1-j}, \text{ где } i = k-1, k, \dots, N-1$$

Здесь β_i – числовые коэффициенты, $f_{i+1-j} = f(x_{i+1-j}, y_{i+1-j})$

Таким образом можно найти новое значение y_{i+1}, используя найденные ранее значения

$y_i, y_{i-1}, \dots, y_{i+1-k}$. Поэтому сначала требуется задание начальных значений $y_0, y_1, \dots, y_{k-1}$.

Их можно вычислить, например, методом Рунге-Кутты, будет описан ниже.

В случае $\beta_0 = 0$ метод Адамса – явный, т.к. значение y_{i+1} в этом случае выражается через найденные значения по явной формуле, написанной выше. Если $\beta_0 \neq 0$, то метод Адамса – неявный.

При $\beta_0 = 0$ из верхнего уравнения получаем

$$y_{i+1} = y_i + h\sum_{j=1}^{k} \beta_i f(x_{i+1-j}, y_{i+1-j}), \text{ где } i = k-1, k, \dots, N-1$$

При k=1, метод совпадает с методом Эйлера.

$$\Delta y = h * f(x, y)$$

$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_n + \Delta y))$$

$$x = a + h$$

Метод Рунге-Кутты для вычислений более 1 начальной точки:

$$k_1 = \varepsilon * f(x_i, y_i)$$

$$k_2 = \varepsilon * f\left(x_i + \frac{\varepsilon}{2}, y_i + \frac{k_1}{2}\right)$$

$$k_3 = \varepsilon * f\left(x_i + \frac{\varepsilon}{2}, y_i + \frac{k_2}{2}\right)$$

$$k_4 = \varepsilon * f(x_i + \varepsilon, y_i + k_3)$$

Далее вычисляем по формуле:

$$y_{i+1} = y_i + \frac{k_1 + 2 * k_2 + 2 * k_3 + k_4}{6},$$

где $i = 0, \dots, 2$

Пусть решение в точке x_i найдено с некоторым h. Выполняется следующий цикл.

Вычисляется погрешность $err = \frac{|y_{i+1} - y\hat{}_{i+1}|}{y_{\hat{}i+1}}$

Если $err \leq tol$, т. е. погрешность меньше или равна заданной допустимой погрешности, то решение в точке x_{i+1} считается найденным, а шаг интегрирования — выполненным успешно. Далее вычисляется новый шаг интегрирования $h = min(\gamma h, hmax)$, где $\gamma \geq 1$, и h принимается в качестве текущего для следующего шага и интегрирование продолжается. Здесь $hmax$ —максимальный шаг (задается пользователем).

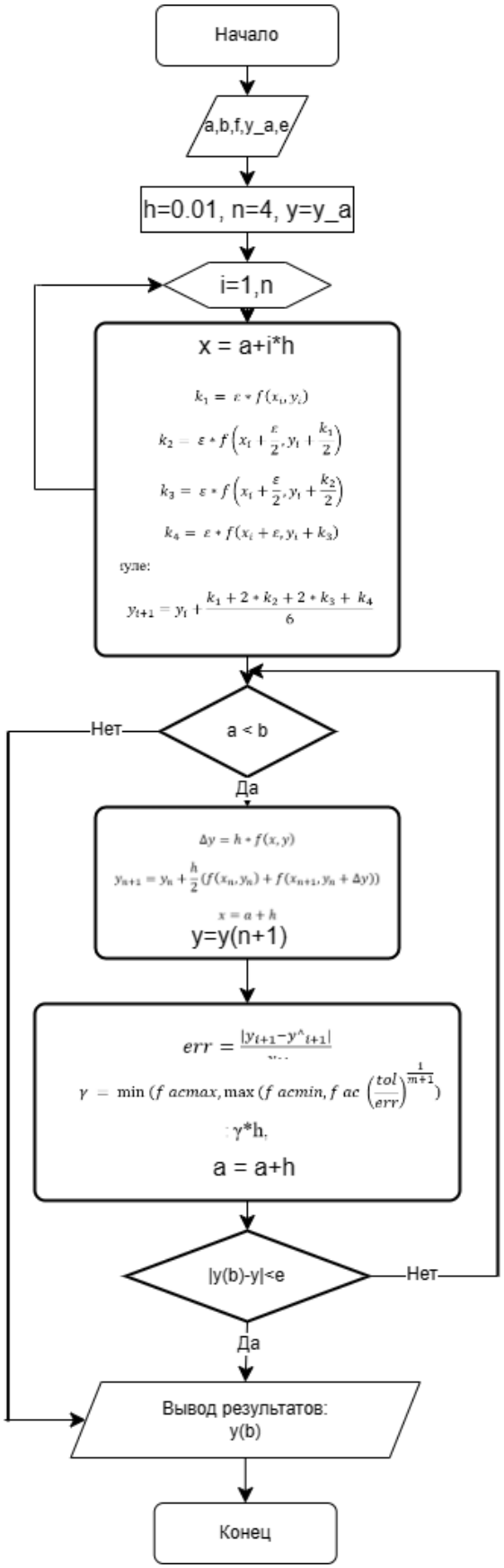
Если $err > tol$, т. е. погрешность оказывается больше желаемой, то полученное решение в точке x_{i+1} отбрасывается как неточное, а шаг интегрирования считается неудачным. Вычисляется новый шаг равный γ*h, где теперь $\gamma < 1$, и вычисления повторяются.

Для выбора γ, как правило, используют формулу:

$$\gamma = \min(f\ acmax, \max(f\ acmin, f\ ac\ \left(\frac{tol}{err}\right)^{\frac{1}{m+1}})$$

$f\ acmax$ ($f\ acmin$) — максимальный коэффициент увеличения (уменьшения) шага ($f\ acmax = 1.5, \ldots, 5$). Эти коэффициенты страхуют от резкого увеличения или уменьшения шага. Для шагов, выполняемых непосредственно после отбрасывания решения, рекомендуется выбирать $f\ acmax = 1$; $f\ ac$ — страховочный коэффициент ($f\ ac = 0.8, \ldots, 0.9$).

Блок-схема численного метод



Код

```
def solveByAdams(f, epsilon, a, y_a, b):
    try:
        func = Result.get_function(f)
        h = 0.01
        y = y_a
        n = 4

        def runge_kutta(x, y, h):
            k1 = h * func(x, y)
            k2 = h * func(x + 0.5 * h, y + 0.5 * k1)
            k3 = h * func(x + 0.5 * h, y + 0.5 * k2)
            k4 = h * func(x + h, y + k3)
            return y + (k1 + 2 * k2 + 2 * k3 + k4) / 6

        points = [(a + i * h, runge_kutta(a + i * h, y, h)) for i in
range(n)]

        while a < b:
            y_pred = y + h * func(a, y)
            y_corrected = y + h * (func(a, y) + func(a + h, y_pred)) / 2
            error = abs(y_corrected - y_pred)
            h *= min(1.0, epsilon / error) ** 0.5
            y = y_corrected

            points.append((a, y))
            points.pop(0)

            a += h

        return points[-1][1]

    except Exception as e:
        print(e)
        return None
```

Метод Адамса.

Примеры работы программы

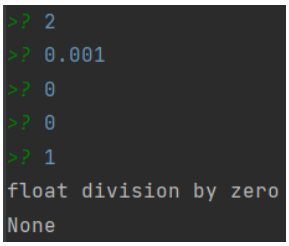
1. Тест для второй функции.

```
>? 2
>? 0.001
>? 0
>? 1
>? 1
1.2840247099352695
```

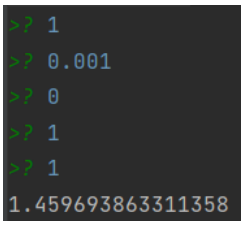
2. Тест для третьей функции.

```
>? 3
>? 0.001
>? 1
>? 2
>? 3
7.919308884890376
```

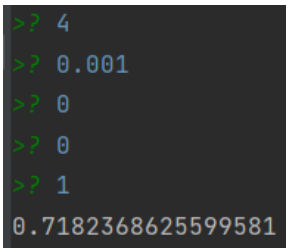
3. Тест случая, в котором произошло деление числа с плавающей точкой на ноль.



4. Тест для первой функции.



5. Тест для четвертой функции.



Вывод

Метод Адамса является численным многошаговым методом решения обыкновенных дифференциальных уравнений, который использует информацию из предыдущих шагов для улучшения точности решения. В зависимости от выбранной функции могут быть граничные случаи, неопределенности, как деление числа с плавающей точкой на ноль (пример есть выше). Сравним метод Адамса с другими методами. Метод Адамса более эффективен, чем методы Эйлера и Рунге-Кутты, поскольку он использует информацию из нескольких предыдущих шагов для улучшения точности. Также метод Эйлера хоть и проще, чем метод Адамса, он может потребовать дополнительные итерации для достижения высокой точности, что может затратить больше времени. Метод Адамса может быть менее точным, чем методы высших порядков (такие как метод Рунге-Кутты 4-го порядка), но он обычно требует меньше вычислений. Метод Адамса отличается от метода Рунге-Кутта тем, что для вычисления очередного значения искомого решения используется не одно, а несколько значений, которые уже вычислены в предыдущих точках. Нужно учитывать, что точность вычислений зависит от размера шагов интегрирования (при больших шагах метод Адамса может быть нестабильным, а также он может потребовать дополнительной памяти).

Метод Адамса может быть использован для решения широкого класса обыкновенных дифференциальных уравнений с любой начальной точкой. Однако, этот метод непригоден для решения уравнений с переменной производной, поскольку он учитывает изменение производной внутри интервала.

Алгоритмическая сложность метода составляет $O(n)$, где n – количество шагов. Анализ численной ошибки в численном методе Адамса. Численная ошибка зависит от выбора шага h и точности вычислений. Чем меньше h, тем меньше ошибка. Метод Адамса может давать большую ошибку, чем методы высших порядков, но он обычно требует меньше вычислений.

В целом, метод Адамса является эффективным и универсальным методом решения обыкновенных дифференциальных уравнений и задач Коши. Но для применения метода нужно более одной начальной точки, для чего потребуется применить другие методы для вычисления этих точек.