Задание 3. Вычисление корней уравнений и определенных интегралов

Постановка задачи

С заданной точностью eps вычислить площадь плоской фигуры, ограниченной тремя кривыми, уравнения которых $y = f_1(x)$, $y = f_2(x)$ и $y = f_3(x)$ определяются вариантом задания.

При решении задачи необходимо:

1) с некоторой точностью eps_1 вычислить абсциссы точек пересечения кривых, используя предусмотренный вариантом задания метод приближенного решения уравнения $f(x) = g(x)$. Отрезки, где программа будет искать точки пересечения и где применим численный метод, определить вручную.
2) представить площадь заданной фигуры как алгебраическую сумму определенных интегралов и вычислить эти интегралы с некоторой точностью eps_2 по квадратурной формуле, предусмотренной вариантом задания.

Величины eps_1 и eps_2 подобрать вручную так, чтобы гарантировалось вычисление площади фигуры с точностью eps.

Варианты задания

Во всех вариантах $eps = 0.001$.

А. Уравнения кривых $y = f_i(x)$

1)	$f_1 = 2^x + 1$	$f_2 = x^5$	$f_3 = (1 - x)/3$
2)	$f_1 = 1.5/(x + 1) + 3$	$f_2 = 2.5x - 9.5$	$f_3 = 5/x \quad (x > 0)$
3)	$f_1 = e^{-x} + 3$	$f_2 = 2x - 2$	$f_3 = 1/x$
4)	$f_1 = e^x + 2$	$f_2 = -1/x$	$f_3 = -2(x + 1)/3$
5)	$f_1 = 0.35x^2 - 0.95x + 2.7$	$f_2 = 3^x + 1$	$f_3 = 1/(x + 2)$
6)	$f_1 = 0.6x + 3$	$f_2 = (x - 2)^3 - 1$	$f_3 = 3/x$

7) $f_1 = \ln x$ $\quad f_2 = -2x + 14 \quad f_3 = 1/(2 - x) + 6$
8) $f_1 = e^x + 2$ $\quad f_2 = -2x + 8 \quad f_3 = -5/x$
9) $f_1 = 3/((x - 1)^2 + 1)$ $\quad f_2 = \sqrt{x + 0.5} \quad f_3 = e^{-x}$
10) $f_1 = 1 + 4/(x^2 + 1)$ $\quad f_2 = x^3 \quad f_3 = 2^{-x}$

Б. Методы приближенного решения уравнений:

1) метод деления отрезка пополам;
2) метод хорд (секущих);
3) метод касательных (Ньютона);
4) комбинированный метод (хорд и касательных).

В. Квадратурные формулы:

1) формула прямоугольников;
2) формула трапеций;
3) формула парабол (Симпсона).

Требования к программе

1. Программа должна состоять из двух модулей: основной модуль, вычисляющий площадь криволинейного треугольника, и модуль, содержащий процедуры, которые реализуют численные методы.

2. Метод приближенного решения уравнения и приближенное вычисление интеграла реализовать в виде процедур:

   а) `root(f,g,a,b)` — вещественная функция, вычисляющая с точностью eps_1 (константа) корень уравнения $f(x) = g(x)$ на отрезке $[a, b]$. Если используется метод касательных или комбинированный метод, то у функции `root` должны быть еще параметры $f1$ и $g1$ — производные функций f и g.

   б) `integral(f,a,b)` — вещественная функция, вычисляющая с точностью eps_2 (константа) величину определенного интеграла от функции $f(x)$ на отрезке $[a, b]$.

3. В основной программе предусмотреть печать абсцисс точек пересечения кривых, количество отрезков разбиения, которое получилось при вычислении интегралов, и печать площади криволинейного треугольника. Печатать только точные цифры результата — используя форматный вывод напечатать только цифры, обеспечиваемые заданной точностью вычислений.

4. Процедуры `root` и `integral` следует предварительно протестировать. Для каждой процедуры нужно составить набор тестовых функций, достаточный для проверки всех существенных вариантов работы `root` и `integral`

5. По итогам работы составить отчёт, который следует предъявить при сдаче задания. Содержание отчета:

1) Постановка задачи — конкретный вариант.
2) Схематичное изображение графиков функций с обозначенными точками пересечения графиков и выделенным криволинейным треугольником; формула выражения площади треугольника через определённые интегралы.
3) Отрезки для поиска точек пересечения графиков, обоснование применимости метода на каждом отрезке.
4) Обоснование выбора eps_1 и eps_2.
5) Набор тестов для функции `root`.
6) Набор тестов для функции `integral`.
7) Результаты выполнения основной программы.

Методические указания

Численное решение уравнений

1. Вместо исходного уравнения $f(x) = g(x)$ будем решать уравнение $F(x) = 0$, где $F(x) = f(x) - g(x)$. Пусть x_0 — точное решение уравнения. Требуется найти точку x', которая близка к точному решению x_0, а именно, $|x' - x_0| < eps_1$. Суть численных методов решения уравнения заключается в построении последовательности точек, сходящейся к точному решению. На каждом шаге вычисляется очередная точка последовательности — очередное приближение. Процесс завершается, когда расстояние от приближения до точного решения стало меньше заданной точности eps_1.

Для корректного применения предложенных методов приближенного решения уравнения $F(x) = 0$ необходимо найти отрезок $[a, b]$, на котором уравнение имеет ровно один корень. Достаточное условие для этого таково: на концах отрезка функция $F(x)$ имеет разные знаки и на всем отрезке производная функции не меняет знак. Кроме того, для методов хорд и касательных, а также комбинированного метода обязательно требуется, чтобы на данном отрезке первая и вторая производные функции не меняли свой знак (не обращались в нуль).

2. В ***методе деления отрезка пополам*** в качестве первого приближения к корню берётся средняя точка c отрезка $[a, b]$. Из двух отрезков $[a, c]$ и $[c, b]$ выбирается тот, на концах которого функция $F(x)$ имеет разные знаки. К выбранному отрезку применяется та же процедура. Процесс деления отрезков прекращается, когда длина очередного отрезка станет меньше требуемой точности eps_1. В качестве приближённого решения можно взять середину отрезка.

В остальных трёх методах следует различать два случая:

- случай 1 — первая и вторая производные функции $F(x)$ имеют одинаковый знак ($F'(x)F''(x) > 0$);
- случай 2 — эти производные имеют разные знаки.

В ***методе хорд*** концы $(a, F(a))$ и $(b, F(b))$ кривой $y = F(x)$ соединяются прямой линией и определяется точка пересечения этой линии с осью абсцисс:

$$c = \frac{aF(b) - bF(a)}{F(b) - F(a)}.$$

Точка c и есть первое приближение к корню. Дальнейшие действия зависят от случая, который имеет место.

Случай 1. Продолжаем вычисления на отрезке $[c, b]$, т. е. полагаем новое значение a равным c, $a_1 = c$. Опять проводим хорду на отрезке $[a_1, b]$, определяем точку пересечения хорды с осью абсцисс (a_2) и т. д. Таким образом строим возрастающую последовательность левых концов отрезков ($a = a_1, a_2, \ldots$) Процесс завершается, когда точное решение отстоит от a_i не дальше, чем на eps_1. Это означает, что $F(x)$ меняет знак на отрезке $[a, a + eps_1]$, т. е. $F(a)F(a + eps_1) < 0$.

Случай 2. Вычисления продолжаются на отрезке $[a, c]$, полагаем теперь $b_1 = c$. Проводим хорду на отрезке $[a, b_1]$, получаем координату пересечения хорды с осью абсцисс — новое значение b. Таким образом получаем убывающую последовательность правых концов отрезков ($b = b_1, b_2, \ldots$) Процесс завершается, когда на отрезке $[b - eps_1, b]$ функция $F(x)$ меняет знак ($F(b - eps_1)F(b) < 0$).

В ***методе касательных***, в зависимости от имеющего место случая, выбирается точка d, проводится касательная к кривой $y = F(x)$ в точке $(d, F(d))$. Точка пересечения касательной с осью абсцисс

$$d_1 = d - \frac{F(d)}{F'(d)}$$

является первым приближением решения уравнения. Далее процесс повторяется: проводится касательная в точке $(d_1, F(d_1))$, вычислив координату пересечения касательной с осью абсцисс, получаем очередное приближение к корню d_2 и т. д. Вычисления заканчиваются, когда приближённое решение оказывается достаточно близко к точному корню.

Случай 1. Касательные проводятся в правом конце отрезка, $d = b$. Последовательность точек $\{d_i\}$ является убывающей, приближение к корню происходит справа; условие окончания вычислений то же, что и в методе хорд: $F(d - eps_1)F(d) < 0$.

Случай 2. Касательные проводятся в левом конце отрезка, $d = a$. Последовательность точек $\{d_i\}$ возрастает, приближение к корню происходит слева; условие окончания вычислений: $F(d)F(d + eps_1) < 0$.

В ***комбинированном методе*** одновременно применяются метод хорд и метод касательных к разным концам отрезка. В случае 1 точка a изменяется по методу хорд, точка b — по методу касательных, в случае 2 — наоборот. Приближение к корню происходит с двух сторон. Критерий окончания — длина очередного отрезка меньше eps_1.

3. При использовании метода хорд, метода касательных или комбинированного метода функция `root` должна самостоятельно распознавать, какой из двух случаев, указанных в п. 2., имеет место при текущем обращении к ней. Это можно сделать проверкой следующих двух условий:

- функция возрастает или убывает;
- график функции расположен выше хорды, соединяющей концы графика, или ниже.

Поскольку производные $F'(x)$ и $F''(x)$ на отрезке $[a, b]$ не меняют знак, для проверки первого условия достаточно сравнить $F(a)$ с 0 (при $F(a) < 0$ функция возрастает). Для проверки же второго условия надо сравнить в какой-то внутренней точке отрезка значения функции и хорды; например, если взять среднюю точку $(a+b)/2$ отрезка, то соотношение

$$F\left(\frac{a+b}{2}\right) > \frac{F(a)+F(b)}{2}$$

означает, что график функции расположен выше хорды (т. е. функция вогнутая). Если функция возрастает и её график расположен ниже хорды или если функция убывает и её график расположен выше хорды, то имеет место случай 1, иначе — случай 2.

4. Для того, чтобы повысить наглядность процедуры `root` и сделать короче выражения, реализующие численный метод, можно описать вложенную функцию, например, `fg(x)`, вычисляющую разность $f(x) - g(x)$.

Численное интегрирование

1. Суть численного интегрирования функции $f(x)$ на отрезке $[a, b]$ состоит в следующем. Отрезок $[a, b]$ делится на n одинаковых отрезков. На каждом из отрезков разбиения вычисление интеграла исходной функции $f(x)$ заменяется на вычисление интеграла более простой функции; приближённое значение интеграла $f(x)$ на отрезке $[a, b]$ получается как сумма интегралов на всех отрезках разбиения. В зависимости от метода, исходная функция $f(x)$ заменяется на константу (метод прямоугольников), прямую (метод трапеций) или параболу (метод Симпсона). Соответствующие квадратурные формулы для приближенного вычисления интеграла I от функции $f(x)$ на отрезке $[a, b]$ имеют следующий вид (n — число отрезков разбиения $[a, b]$):

Формула прямоугольников:

$$I \approx I_n = h(f_0 + f_1 + \cdots + f_{n-1}),$$

где $f_i = f(a + (i + 0.5)h)$, $h = \frac{b-a}{n}$

Формула трапеций:

$$I \approx I_n = h(0.5f_0 + f_1 + f_2 + \cdots + f_{n-1} + 0.5f_n),$$

где $f_i = f(a + ih)$, $h = \frac{b-a}{n}$

Формула Симпсона (n — чётное):

$$I \approx I_n = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 4f_{n-3} + 2f_{n-2} + 4f_{n-1} + f_n),$$

где $f_i = f(a + ih)$, $h = \frac{b-a}{n}$

2. Для достижения требуемой точности обычно используется следующий метод: берётся некоторое начальное число разбиений n_0 (1, 2 или 3) и последовательно вычисляются значения I_n при n, равном $2n_0$, $4n_0$, $8n_0$ и т. д. Известно *правило Рунге*

$$|I - I_n| \approx p|I_n - I_{2n}|$$

(для формул прямоугольников и трапеций $p = 1/3$, для формулы Симпсона $p = 1/15$). Согласно этому правилу, когда на очередном шаге величина $p|I_n - I_{2n}|$ окажется меньше eps_2, в качестве приближенного значения для I можно взять I_n или, что лучше, I_{2n}.

3. При реализации функции `integral` следует учитывать, что в формулах трапеций и Симпсона в сумму I_{2n} входят значения f_i, вычисленные ранее для суммы I_n, поэтому их не следует перевычислять заново.

Отладка функций root и integral

1. В общем случае процесс отладки программы заключается в тестировании программы с целью выявления ошибок и в исправлении найденных ошибок. Подготавливается набор тестовых данных, таких, что правильный результат известен заранее. Программа применяется к тестовым данным, получившийся результат сравнивается с правильным ответом. Тестовые данные должны быть настолько простые, чтобы несложно было вручную получить ответ. Данных в тестовом наборе должно быть достаточное количество для проверки всех существенных вариантов работы тестируемой программы.

В задании требуется отладить не программу, а две процедуры. Для отладки процедуры нужно написать вспомогательную программу, которая вызывает процедуру для тестовых данных и печатает результат. Тестовыми данными являются функция и отрезок (параметры процедуры).

При отладке реализации численного метода нужно проверить, как точность метода влияет на вычисленный результат. Для этого нужно прогнать отладочную программу с разными значениями точности для одних и тех же тестовых данных и сравнить, на сколько вычисленные результаты отличаются от точного решения.

2. При отладке `root(f,g,a,b)` тест состоит из пары функций f и g и отрезка для нахождения корня. Удобно взять $g \equiv 0$. Отрезок

должен быть таким, чтобы разность функций на нём удовлетворяла условиям применимости численного метода (см. п. 1. раздела Численное решение уравнений). Набор тестов должен проверять все существенные случаи работы процедуры `root`. Для метода деления отрезка пополам по крайней мере один тест должен проверять случай, когда середина отрезка является корнем. Для остальных методов набор должен содержать тесты, соответствующие четырём случаям — всем возможным комбинациям возрастания/убывания и выпуклости/вогнутости функции. При отладке методов хорд и касательных процедура должна печатать, с какой стороны идёт приближение к корню; для комбинированного метода нужно печатать, к какому концу отрезка применяется метод хорд, а к какому — метод касательных. Для сравнения результатов вычислений, выполненных с разной точностью, во время отладки нужно печатать все знаки результата, без форматирования.

3. При отладке `integral(f,a,b)` тест состоит из функции f и отрезка интегрирования. Процедура должна печатать количество точек разбиения. Вообще говоря, чем лучше точность, тем больше отрезков разбиения получится. Однако, если интегрируемая функция в силу метода заменяется на саму себя, количество отрезков разбиения не зависит от точности (т. к. приближённое значение интеграла равно точному). В тестовый набор, помимо других тестов, должен входить тест, проверяющий эту ситуацию.

Функциональные параметры в среде Турбо Паскаль

В функциях `root` и `integral` используются параметры-функции (f, g и др.). В языке Турбо Паскаль такие параметры передаются следующим образом.

В разделе типов необходимо описать так называемый функциональный тип:

```
type TF = function (x: Real): Real;
```

(вместо `TF` и `x` можно использовать любые другие имена). Все вещественные функции от одного вещественного аргумента, передаваемого по значению, описанные в программе, автоматически включаются в функциональный тип `TF`. Имя данного типа нужно указать в спецификации формального параметра-функции в заголовках `root` и `integral`:

```
function root(f,g: TF; a,b: Real): Real;
```

При обращении к функциям `root` и `integral` указываются имена фактических параметров-функций, например:

```
root(f1, f2, -0.1, 3.5)
```

Замечание: стандартные функции нельзя передавать в качестве фактических параметров.

Перед описанием функций, передаваемых как фактические параметры (`f1`, `f2` и др.), необходимо разместить директиву транслятору `{$F+}`, например:

```
{$F+}
function f1(x:Real):Real; begin f1:=ln(x)  end;
function f2(x:Real):Real; begin f2:=2*x+14 end;
```

Директива `{$F+}` делает вызовы описанных вслед за ней процедур и функций дальними (по умолчанию действует директива `{$F-}` и все описываемые процедуры и функции объявляются ближними). Описание функционального типа также следует производить после директивы `{$F+}`, в противном случае дальние функции нельзя будет передавать в качестве аргументов такого типа.