Язык программирования Фортран

Fortran 90, 95, 2003, 2008

Практика в комп.классе

ФТФ

Модуль № 1 «Основные свойства Фортрана»

Оглавление

Тема № 1. Программы линейной структуры

На этом занятии необходимо познакомиться с основами работы в среде *Compaq Visual Fortran 6.1* и *PGI Visual Fortran 2010*. Следует разобрать такое понятие, как *проект* и научиться основам управления проектами. Требуется понять структуру программы на Фортране, изучить основы ввода – вывода, научиться составлять программы линейной структуры.

Список заданий

Задание № 1

Даны действительные числа x и y. Получить:

$$\frac{|x|+|y|}{1+|x\cdot y|}$$

Значения переменных x и y получить с клавиатуры, значение вычисленного выражения вывести на экран.

Задание № 2

Даны два вещественных числа x_1 и x_2. Получить коэффициенты приведенного квадратного уравнения с корнями x_1 и x_2. Расчетные формулы получить либо вручную, либо при помощи системы *Mathematica*. Значения переменных x_1 и x_2 находятся в текстовом файле **`Coeffs.dat`**. Рассмотреть случаи, когда оба вещественных числа указаны на одной строке файла и когда каждое число расположено на отдельной строке. Результаты вычисления вывести на экран и в файл **`Res.dat`**.

Задание № 3

Треугольник задан координатами своих вершин на плоскости. Найти периметр и площадь треугольника. Значение координат вершин треугольника ввести с клавиатуры, результаты расчета вывести на экран и в файл **`Triangle.res`**. Проверить полученные результаты на собственных тестовых примерах.

Задание № 4

Заданы действительные числа c и d. Вычислить значение выражения:

$$\left|\frac{\sin^3|c\cdot x_1^3+d\cdot x_2^2-c\cdot d|}{\sqrt{\left(c\cdot x_1^3+d\cdot x_2^2-x_1\right)^2+\pi}}\right|+\mathrm{tg}\left(c\cdot x_1^3+d\cdot x_2^2-x_1\right),$$

где x_1 — больший, а x_2 — меньший корни квадратного уравнения $x^2-3\cdot x-|c\cdot d|=0$. Значения переменных c и d получить с клавиатуры, результат вычислений вывести на экран. Проверить полученные результаты с помощью системы *Mathematica* и с помощью тестовых примеров.

Значения для проверки:

входные значения:	$c=0.3$,	$d=0.4$.
промежуточные результаты:	$x_1=2.95945$,	$x_2=0.040548$.
финальный результат:	-9.32363	

Задание № 5

Даны действительные числа x, y и z. Вычислить значения переменных a и b, если:

a). $$a=\frac{3+e^{y-1}}{1+x^2|y-\operatorname{tg}(z)|},\qquad b=1+|y-x|+\frac{(y-x)^2}{2}+\frac{|y-x|^3}{3};$$

b). $$a=(1+y)\frac{x+\dfrac{y}{x^2+4}}{e^{-x-2}+\dfrac{1}{x^2+4}},\qquad b=\frac{1+\cos(y-2)}{\dfrac{x^4}{2}+\sin^2(z)};$$

c). $$a=\frac{2\cos\left(x-\dfrac{\pi}{6}\right)}{\dfrac{1}{2}+\sin^2(y)},\qquad b=1+\frac{z^2}{3+\dfrac{z^2}{5}};$$

d). $$a=\ln\left|\left(y-\sqrt{|x|}\right)\left(x-\frac{y}{z+\dfrac{x^2}{4}}\right)\right|,\qquad b=x-\frac{x^2}{3!}+\frac{x^5}{5!};$$

Значения переменных x, y и z получить с клавиатуры, значение вычисленного выражения вывести на экран. Проверить полученные результаты с помощью системы *Mathematica* и с помощью тестовых примеров.

Значения для проверки:

входные значения:	$x=1.3$	$y=2.4$	$z=2.7$
финальный результат:	a).	$a=1.20501$	$b=3.14867$
	b).	$b=27.5318$	$b=1.19268$
	c).	$a=1.49216$	$b=2.63526$
	d).	$a=-0.401296$	$b=1.04927$

Рекомендации по выполнению

Некоторые встроенные функции

Некоторые встроенные функции, требуемые для решения заданий, приведены в таблице:

Математическая запись	Запись на Фортране
$\sqrt{x}$	`sqrt(x)`
$\|x\|$	`abs(x)`
e^x	`exp(x)`
$\ln(x)$	`log(x)`
$\sin(x)$	`sin(x)`
$\cos(x)$	`cos(x)`
$\operatorname{tg}(x)$	`tan(x)`

Замечания:

Данные функции могут работать с аргументами как вещественного, так и комплексного типа. При этом результат зависит от типа аргумента. Аргументы тригонометрических функций задаются в радианах.

Работа в среде Developer Studio

Для удобства работы, разработка приложений в интегрированной среде Developer Studio организована с помощью проектов. *Проект* (*project*) — это набор взаимосвязанных исходных файлов, компиляция и компоновка которых позволяет создавать исполняемую программу нужного типа или библиотеку. Исходные файлы проекта обычно хранятся в отдельном подкаталоге. Кроме исходных файлов с текстами программ проект обычно включает набор служебных файлов. Например, в проекте всегда есть *файл проекта* (*project file*, файл имя которого совпадает с названием проекта и расширением имени файла **DSP**), *файл рабочего пространства* (*workspace file*, файл, имя которого совпадает с названием проекта и расширением имени файла **DSW**). Файл проекта — это текстовый файл, в котором хранятся параметры компилятора и компоновщика, а также отражаются все взаимосвязи между исходными модулями. Файл рабочего пространства — это текстовый файл, содержащий список всех проектов данной группы. Для того, чтобы начать работу с существующим проектом нужно выбрать (открыть в среде разработчика) соответствующий файл рабочего пространства. Некоторые другие типы служебных файлов, входящих в состав проекта, приведены в **Табл. 1.1**.

Таблица 1.1. Некоторые типы служебных файлов, входящие в состав проекта

DEP	Файл зависимостей
DSP	Файл проекта
DSW	Файл рабочего пространства
OPT	Файл конфигурации рабочего пространства
PLG	Файл журнала сборки

Для того, чтобы создать новый проект необходимо в строке меню главного окна программы выбрать *File | New*. В результате на экран будет выведено диалоговое окно *New* с активной вкладкой *Projects* (см. **Рис. 1.1**).

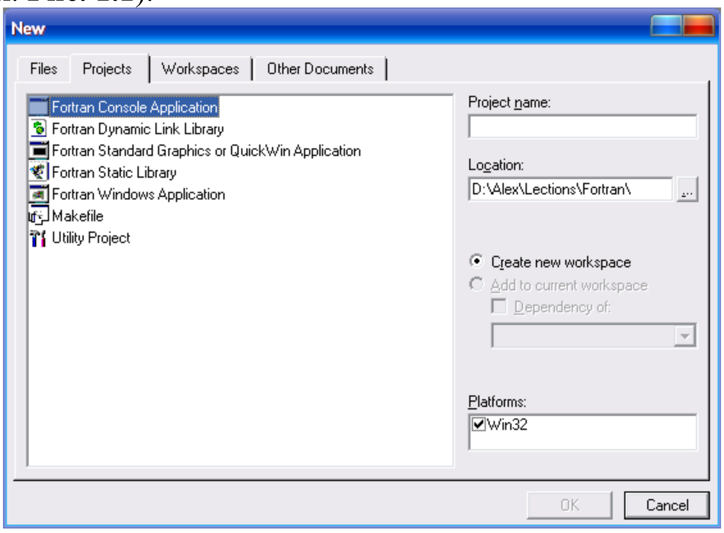

Рисунок 1.1. Диалоговое окно *New*. Первый шаг мастера создания нового проекта

В данном диалоговом окне сначала нужно выбрать тип проекта (*Fortran Console Application — Консольное приложение на Фортране*). Затем в текстовом поле *Location* следует указать месторасположение создаваемого проекта. Выбранное место в файловой системе компьютера можно вручную записать в данном поле, а можно нажать кнопку с тремя точками, расположенную справа от него. На экран будет выведено стандартное диалоговое окно выбора каталога. В нем нужно мышью выбрать нужный логический диск компьютера, нужную папку и нажать кнопку *OK*. Полный путь к выбранной папке будет автоматически занесен в текстовое поле *Location*. Далее, в текстовом поле *Project Name* нужно указать имя проекта. Следует отметить, что имя проекта и полный путь к нему не должен содержать кириллических символов. После того, как указана вся необходимая для создания проекта информация нужно нажать кнопку *OK*. На экран будет выведено следующее диалоговое окно создания консольного приложения Фортрана (см. **Рис. 1.2**).

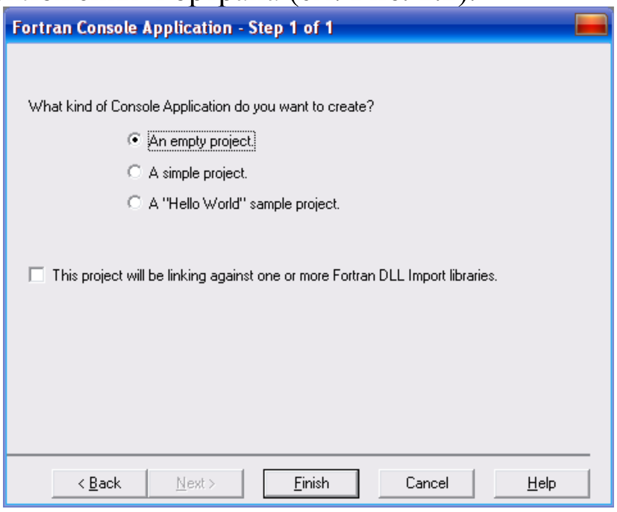

Рисунок 1.2. Диалоговое окно *Fortran Console Application - Step 1*. Второй шаг мастера создания нового проекта

В данном диалоговом окне нужно уточнить тип консольного проекта, который должен сделать мастер. Для этого следует выбрать первый пункт *An empty project* (*Пустой проект*) и нажать кнопку *Finish*. В результате на экран будет выведено диалоговое окно *New Project*, содержащее общую информацию о создаваемом проекте (см. **Рис. 1.3**).

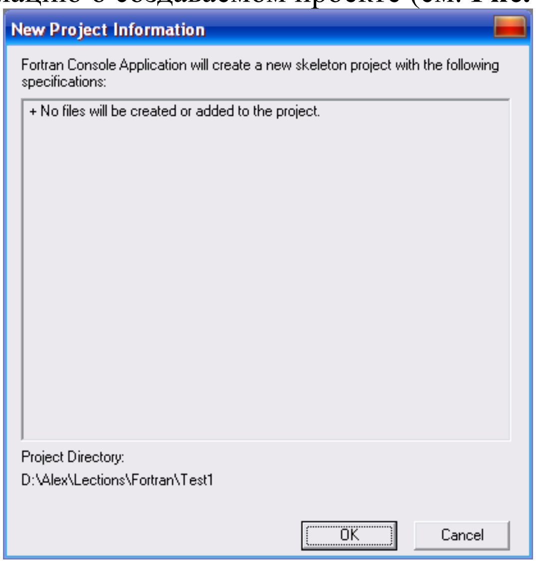

Рисунок 1.3. Диалоговое окно *New Project Information*. Последний шаг мастера создания нового проекта

Необходимо проверить суммарную информацию о проекте, и если все в порядке — нажать кнопку *OK*. В указанном месте файловой системы компьютера будет создана папка проекта и необходимые служебные файлы.

Далее необходимо добавить к проекту файлы с исходным текстом программы. Для этого в строке меню главного окна программы следует выбрать либо пункт

Project | Add To Project | New для создания и автоматического добавления нового файла в проект, либо *Project | Add To Project | Files* для добавления к проекту уже существующего файла с исходным кодом программы. В последнем случае на экран будет выведено стандартное диалоговое окно выбора файла *Insert Files into Project*, в котором нужно указать добавляемый к проекту файл. Если создается новый файл, то на экран выводится диалоговое окно *New*, открытое на вкладке создания файлов *Files* (см. **Рис. 1.4**).

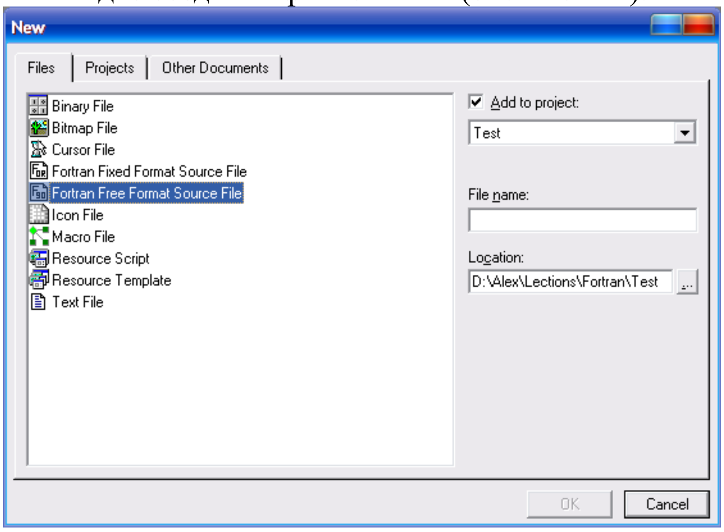

Рисунок 1.4. Диалоговое окно *New*. Создание нового файла с исходным текстом программы и добавление его в активный проект

В данном диалоговом окне нужно сначала выбрать нужный тип файла *Fortran Free Format Source File* (*Исходный файл в свободном формате Фортрана*) или *Fortran Fixed Format Source File* (*Исходный файл в фиксированном формате Фортрана*). Далее, следует проверить, что установлен флажок *Add to project,* в комбинированном списке *File Name* выбран нужный проект, к которому следует добавить нужный файл и в поле *Location* указано требуемое местоположение файла. По умолчанию файлы размещаются в папке проекта. Затем, в текстовом поле *File Name* нужно указать требуемое имя файла и нажать кнопку *OK*. Будет создан файл с исходным тестом программы и автоматически включен в активный проект (см. **Рис. 1.5**).

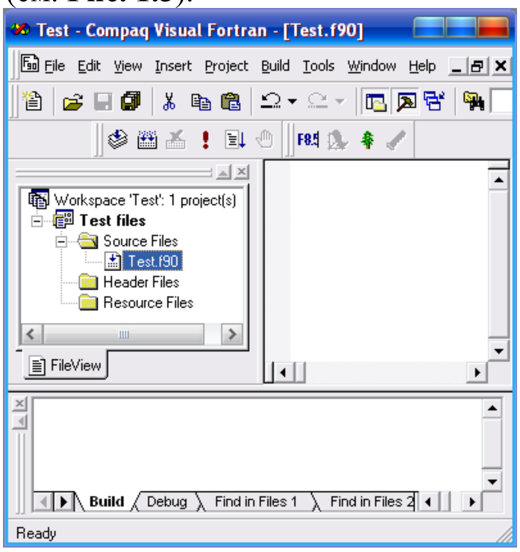

Рисунок 1.5. Созданный и включенный в состав проекта файл. Структура проекта отображается в окне проекта

Для того, чтобы исключить файл из состава проекта нужно выделить мышью его имя в окне проекта и нажать клавишу `Delete` на клавиатуре. Выделенный пользователем файл останется на диске, но будет исключен из состава проекта.

Для того, чтобы создать исполняемый файл программы следует вначале указать требуемую конфигурацию сборки. Для этого в строке меню главного окна программы нужно выбрать пункт *Build | Set Active Configuration*. На экран будет выведено диалоговое окно *Set Active Project Configuration* со списком возможных конфигураций (см. **Рис. 1.6**).

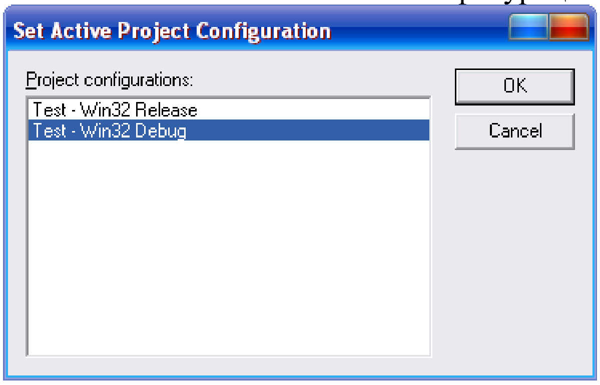

Рисунок 1.6. Диалоговое окно *Set Active Project Configuration*. Выбор требуемой конфигурации проекта

Конфигурация *Debug* предназначена для создания отладочной версии программы, а конфигурация *Release* — для финальной версии программы. После выбора нужной конфигурации следует откомпилировать файл с исходным текстом программы. Для этого нужно убедиться, что в данный момент окно с требуемым файлом является активным. Если это не так, то нужно щелкнуть на нем мышью. Затем можно компилировать требуемый исходный файл. Для этого можно в строке меню главного окна программы выбрать *Build | Compile*. Процесс компиляции можно запустить с клавиатуры, нажав комбинацию клавиш *Ctrl + F7*, а либо с помощью мыши, щелкнув на значке  панели инструментов *Build MiniBar*. Результаты процесса компиляции будут отображены в окне вывода сообщений компилятора и компоновщика, расположенного вдоль нижнего края главного окна программы. Если во время компиляции были обнаружены ошибки, следует исправить первую и заново перекомпилировать программу. Такой процесс исправления следует продолжать до тех пор, пока компилятор не перестанет выдавать сообщения об ошибках. Когда компиляция исходного файла программы успешно завершилась, можно переходить к этапу сборки. Для этого нужно вызвать компоновщик. Сделать это можно с помощью пункта меню *Build | Build*, либо с помощью клавиатуры (клавиша **`F7`**), либо с помощью значка  панели инструментов *Build MiniBar*. Если на этапе компоновки обнаружены ошибки, то соответствующие сообщения будут выведены в окне вывода сообщений компилятора и компоновщика. После успешной компиляции программы в подпапке *Debug* или *Release* (в зависимости от выбранной конфигурации) папки проекта будет сохранена соответствующая версия исполняемой программы (файл с именем проекта и расширением имени файла **`EXE`**).

Процесс полной перекомпиляции и пересборки проекта можно запустить с помощью пункта главного меню программы *Build | Rebuild All*, либо с помощью значка панели инструментов *Build MiniBar*. Запустить на выполнение созданный исполняемый файл программы можно из среды Developer Studio. Для этого нужно выбрать пункт *Build | Execute*, либо нажать на клавиатуре комбинацию клавиш *Ctrl + F5*, либо щелкнув мышью на значке  панели инструментов *Build MiniBar*.

Работа в среде *PGI Visual Fortran 2010* аналогична работе с более ранней версии Developer Studio. Однако есть некоторые отличия. Так, например то, что ранее называлось *Workspace* (*рабочее пространство*) в новой версии студии разработчика называется *Solution* (*решение*). Следует отметить, что между этими понятиями есть различия, которые мы рассматривать не будем. Для создания нового проекта в среде следует *PGI Visual Fortran 2010* следует либо на *Стартовой Странице* (*Start Page*) выбрать команду *New Project*, либо в пункте меню *File* выбрать команду *New* | *Project* (см. **Рис. 1.7**).

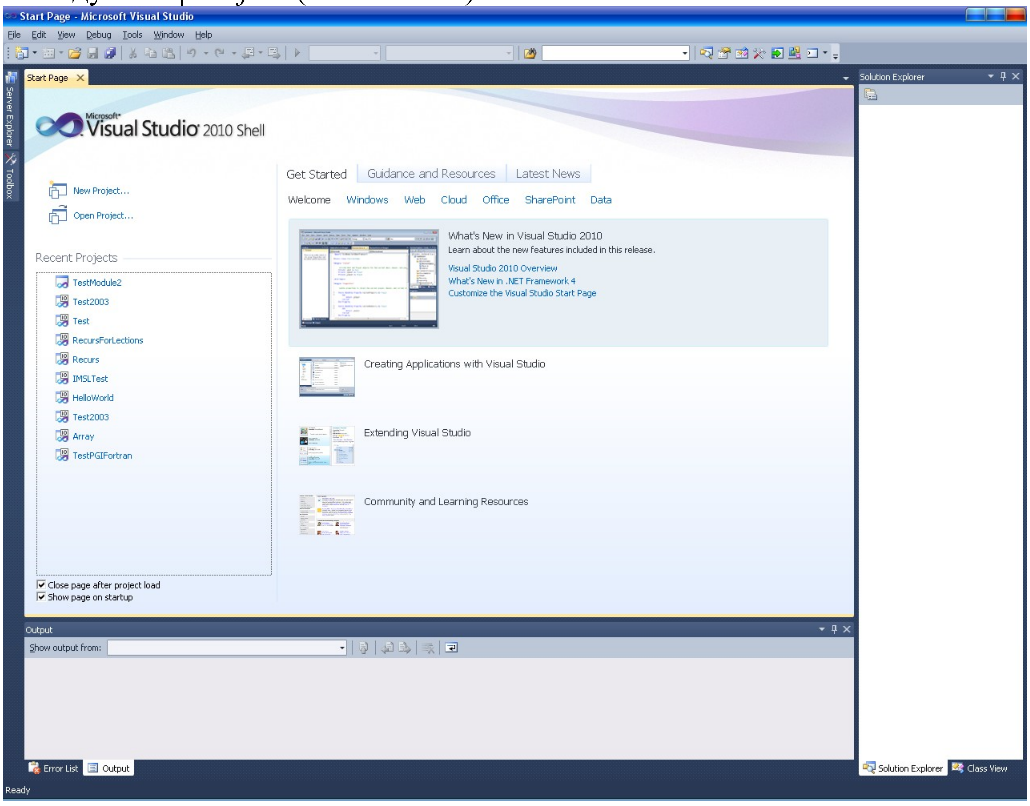

Рисунок 1.7. *Стартовая страница* (*Start Page*) *PGI Visual Fortran 2010*

На экран будет выведено окно мастера создания нового проекта *New Project* (см. **Рис. 1.8**).

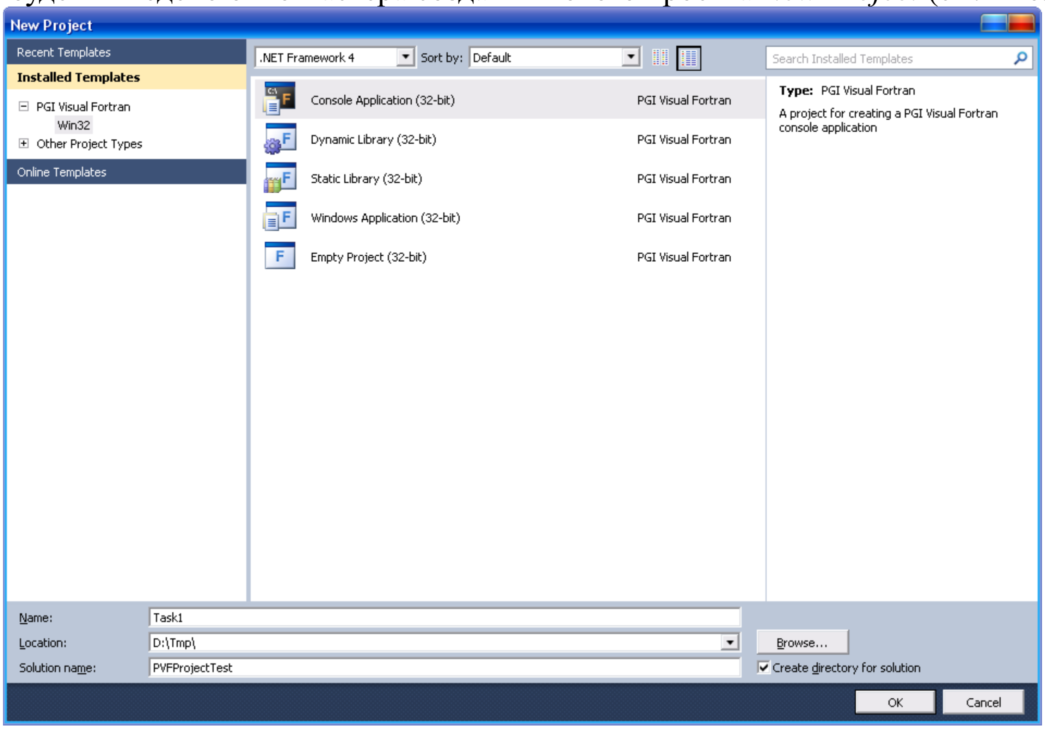

Рисунок 1.8. Окно создания нового проекта *New Project*

В данном окне следует выбрать версию *Net Framework*, для которой планируется создать приложение (Рекомендуется оставить значение, указанное в этом поле по умолчанию:

.Net Framework 4). В списке *Installed Templates* нужно выбрать шаблон проекта *PGI Visual Fortran* | *Win 32* и в списке доступных проектов указать нужный; в нашем случае это *Console Application* (*32-bit*) (*Консольное 32-битное Приложение*). Далее в поле *Location* нужно указать месторасположение проекта. Его можно вписать вручную, либо выбрать с помощью стандартного диалога выбора папки, который вызывается нажатием кнопки *Browse...*, расположенной сразу справа от этого поля. Затем следует указать название решения (поле *Solution name*) и названия проекта (поле *Name*). Эти имена могут быть как одинаковыми, так и разными. Далее следует убедиться, что установлен флажок *Create directory for solution*. После того, как все настройки сделаны нужно нажать на кнопку *OK*. Будет создан проект требуемого типа с заданным именем в указанном места файловой системы (см. **Рис. 1.9**).

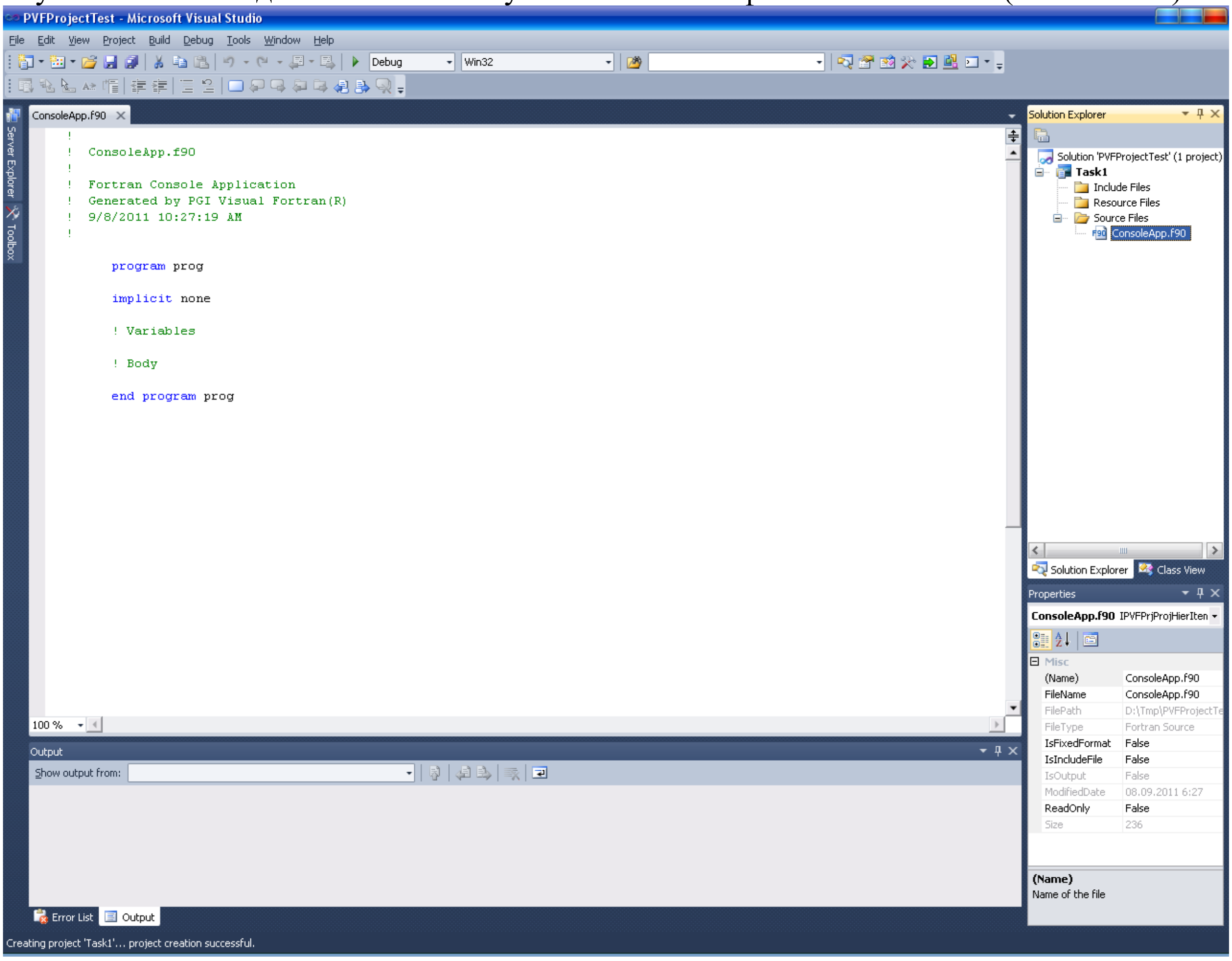

Рисунок 1.9. Окно среды разработчика с новым проектом

В проект будет автоматически добавлен файл в свободном формате **ConsoleApp.f90** с шаблоном головной программы на *Фортране*. Для того, чтобы дать файлу другое имя нужно в окне *Solution Explorer* (*Обозреватель решения*) выбрать мышью нужный файл и щелчком правой кнопки мыши вызвать контекстное меню (см. **Рис. 1.10**).

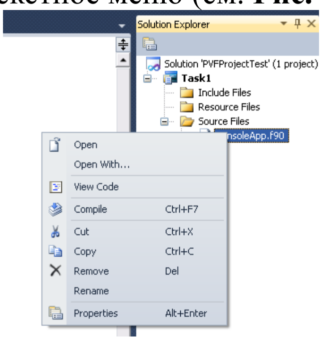

Рисунок 1.10. Контекстное меню файла

В вызванном контекстном меню следует выбрать команду *Rename* (*Переименовать*) и указать требуемое имя файла. Если требуется убрать файл из проекта, нужно выбрать команду контекстного меню *Remove* (*Удалить*), либо как и раньше, выбрать файл мышью и нажать на клавиатуре кнопку **Delete**. На экран будет выведено диалоговое окно (см. **Рис. 1.11**), в котором нужно уточнить, что же следует сделать: просто удалить файл из проекта (кнопка *Remove*) либо удалить файл как из проекта, так и из файловой системы компьютера (кнопка *Delete*).

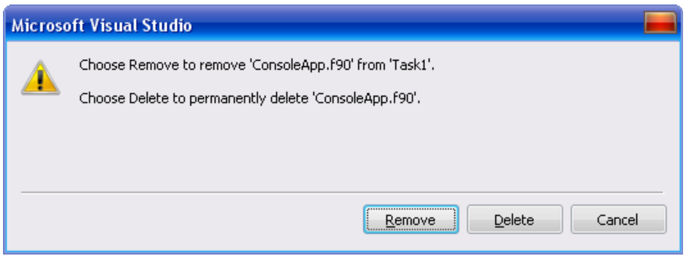

Рисунок 1.11. Диалоговое окно удаления файла из проекта

Для того, чтобы добавить файлы и исходным тестом программы к проекту следует выбрать пункт главного меню программы *Project | Add New Item...* для создания нового файла и добавления его к проекту, либо пункт *Project | Add Existing Item...* для подключения уже существующего файла в вашему проекту. Такие же команды можно выбрать и из контекстного меню проекта, которое можно вызвать из окна *Solution Explorer* (*Обозреватель проектов*) (см. **Рис. 1.12**).

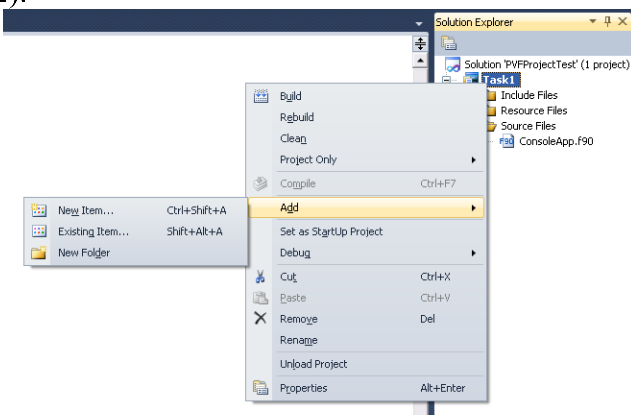

Рисунок 1.12. Диалоговое окно удаления файла из проекта

Команда добавления существующего файла вызовет на экран стандартное диалоговое окно системы выбора файла, а команда создания нового файла вызовет на экран окно создания нового файла *Add New Item* (см. **Рис. 1.13**). В данном окне следует выбрать тип файла (в основном мы будет работать с файлами, записанными в свободном формате — *Free-Format Fortran Source File* (*.f90*)) и в поле *Name* (*Имя*) указать требуемое имя файла. После этого следует нажать кнопку *Add* (*Добавить*) и требуемый файл будет создан и автоматически подключен к проекту.

Перед компиляцией файла следует выбрать требуемую конфигурацию с помощью раскрывающегося списка на главной панели инструментов:  . Для компиляции файла нужно либо выбрать команду главного меню *Build | Compile*. При этом текстовый курсор должен находиться в том файле, который нужно откомпилировать. Кроме того, эту команду можно вызвать из контекстного меню компилируемого файла, которое можно вызвать из окна *Solution Explorer* (см. **Рис. 1.10**). Для сборки проекта нужно воспользоваться командой *Build | Build project name*. Все остальные команды аналогичны командам старой среды Developer Studio.

Для запуска успешно созданной программы на выполнение нужно выбрать пункт меню *Debug | Start Without Debugging* (клавиатурная комбинация **Ctrl+F5**).

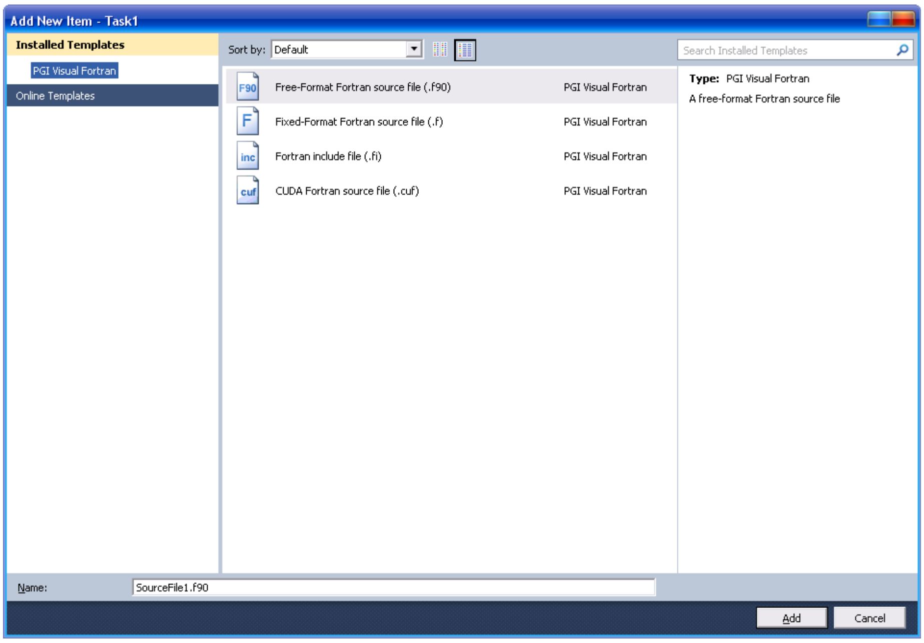

Рисунок 1.13. Диалоговое окно создания нового файла

Для закрытия проекта нужно выбрать команду меню *File | Close Solution*. Открыть проект можно командой меню *File | Open | Project / Solution*, либо команду *Open Project ...* со стартовой станицы среды разработки (см. **Рис. 1.7**).

Тема № 2. Программы разветвляющейся структуры

Это занятие посвящено программам разветвляющейся структуры. На занятии необходимо научиться применять различные операторы ветвления и другие операторы *Фортрана*, которые предназначены для выбора пути выполнения программы.

Список заданий

Задание № 1

Даны вещественные числа x, y, z. Вычислить:

a) $\max(x+y+z, x\cdot y\cdot z)$

b) $\min^2(x+y+z/2, x\cdot y\cdot z)+1$

Замечание: Вычисления провести не используя встроенные функции `min` и `max`.

Задание № 2

Если сумма трех различных вещественных чисел x, y, z меньше единицы, то наименьшее из этих трех чисел заменить полусуммой двух других; в противном случае заменить меньшее из x и y полусуммой двух оставшихся значений.

Задание № 3

Даны действительные числа x, y. Если x и y отрицательны, то каждое значение заменить его модулем; если отрицательно только одно из них, то оба значения увеличить на 0.5; если оба значения неотрицательны и ни одно из них не принадлежит отрезку $[0.5, 2.0]$, то оба значения уменьшить в 10 раз; в остальных случаях x и y оставить без изменений.

Задание № 4

Дано вещественное число h. Найти корни квадратного уравнения $a\cdot x^2+b\cdot x+c=0$, если:

$$a=\sqrt{\frac{|\sin(8h)|+17}{(1-\sin(4h)\cos(h^2+18))^2}}, \qquad b=1-\sqrt{\frac{3}{3+|\mathrm{tg}(a h^2)-\sin(a h)|}},$$
$$c=a h^2\sin(b h)+b h^3\cos(a h).$$

Полученный результат проверить с помощью тестовых примеров. Программа должна находить и вещественные, и комплексные корни уравнения. При реализации программы не следует использовать переменные комплексного типа данных. Тестовые примеры приведены в разделе *Рекомендации по выполнению* этого занятия.

Задание № 5

Дано вещественное число a. Вычислить значение $f(a)$, если функция $f(x)$ задана формулой:

a) $$f(x)=\begin{cases} 0 & \text{при } x\leqslant 0, \\ x & \text{при } 0<x\leqslant 1, \\ x^4 & \text{в остальных случаях;} \end{cases}$$

b) $f(x)=\begin{cases} 0 & \text{при } x\leqslant 0, \\ x^2-x & \text{при } 0<x\leqslant 3 \text{ и } 10<x\leqslant 13, \\ x^2-\sin(\pi x^2) & \text{в остальных случаях;} \end{cases}$

Задание № 6

Заданы три числа a, b и c. Определить, могут ли они быть сторонами треугольника, и если да, то определить его тип: тупоугольный, остроугольный или прямоугольный.

Замечание: Для решения этой задачи можно при помощи теоремы косинусов $a^2=b^2+c^2-2bc\cos(\alpha)$ определить косинус соответствующего угла треугольника. Затем, по знаку косинуса, можно определить вид угла.

Задание № 7

Точка на плоскости задана своими координатами (x, y). Написать программу, которая проверяет, принадлежит ли точка заштрихованной области (см. **Рис. 2.1**).

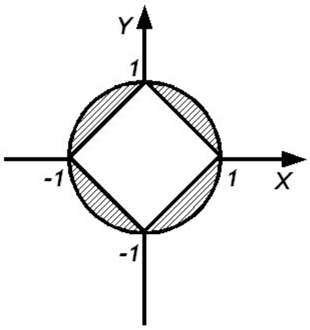

Рисунок 2.1. Область для Задания № 6

Задание № 8

Треугольник на плоскости задан координатами вершин. Принадлежит ли точка (x, y) треугольнику?

Замечание: Для решения этой задачи можно воспользоваться тем фактом, что две точки (x_1, y_1) и (x_2, y_2), не лежащие на кривой $F(x, y)=0$, разбивающей координатную плоскость на две части, принадлежат одной и той же части плоскости, если $F(x_1, y_1)$ и $F(x_2, y_2)$ — числа одного знака.

Задание № 9

Заданы координаты трех точек на плоскости (x_1, y_1), (x_2, y_2), (x_3, y_3). Выяснить, находится ли точка с координатами (x, y) внутри окружности, проведенной через заданные точки? Расчетные формулы получить либо вручную, либо при помощи системы *Mathematica*. Если расчетные формулы получены с помощью системы *Mathematica*, то в формат *Фортрана* их можно преобразовать при помощи функции `FortranForm[ ]`.

Задание № 10

Для заданного числа n $(0\leqslant n\leqslant 100)$, рассматриваемого как возраст человека, вывести на экран правильно сформированную фразу: *21 god*, *32 goda*, *18 let* и т.д.

Рекомендации по выполнению

Задание № 3

Тестовые примеры:

h	a	b	c	x_1	x_2
-1.5	4.3959	0.112484	-2.02238	-0.691193	0.665604
-0.5	2.40697	0.194176	-0.0670465	-0.21204	0.131368
0.0	4.12311	1.1×10^{-16}	0.0	-2.6×10^{-17}	0.0
0.5	16.9031	0.140765	0.287276	-0.00416389 $-0.1303\cdot i$	-0.00416389 $+0.1303\cdot i$
1.5	3.99827	0.0279651	0.46781	-0.00349715 $-0.342039\cdot i$	-0.00349715 $+0.342039\cdot i$

Задание № 8

При выполнении задания удобно воспользоваться тем фактом, что уравнением прямой, проходящей через две точки (x_1, y_1) и (x_2, y_2), является уравнение:

$$(x-x_1)\cdot(y_2-y_1)-(y-y_1)\cdot(x_2-x_1)=0.$$

Задание № 9

Приведем основные характеристики окружности, проходящей через три точки на плоскости: точки (x_1, y_1), (x_2, y_2) и (x_3, y_3). Радиус этой окружности можно вычислить по формуле:

$$r^2=\frac{\left((x_1-x_2)^2+(y_1-y_2)^2\right)\left((x_1-x_3)^2+(y_1-y_3)^2\right)\left((x_2-x_3)^2+(y_2-y_3)^2\right)}{4\,(x_1(y_3-y_2)+x_2(y_1-y_3)+x_3(y_2-y_1))^2}$$

Координаты центра вычисляются по формулам:

$$x_0=\frac{(y_2-y_3)\left(x_1^2+(y_1-y_2)(y_1-y_3)\right)+x_2^2(y_3-y_1)+x_3^2(y_1-y_2)}{2\,(x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2))}$$

$$y_0=\frac{x_1^2(x_3-x_2)+x_1\left(x_2^2-x_3^2+y_2^2-y_3^2\right)-x_2^2x_3+x_2\left(x_3^2-y_1^2+y_3^2\right)+x_3(y_1-y_2)(y_1+y_2)}{2(x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2))}$$

Задание № 10

Для вычисления остатка от деления двух натуральных чисел a и b можно воспользоваться формулой:

$$a-\left(\frac{a}{b}\right)\cdot b$$

Тема № 3. Программы циклической структуры

Это занятие посвящено программам циклической структуры. На занятии необходимо научиться применять различные операторы организации цикла *Фортрана* и познакомится с основными вычислительными алгоритмами.

Список заданий

Задание № 1

Дано натуральное число n. Получить n вещественных чисел $a_1, a_2, \ldots a_n$. Вычислить:

a) $a_1+a_2+\ldots+a_n$

b) $|a_1|+|a_2|+\ldots+|a_n|$

c) $\left(\sqrt{|a_1|}-a_1\right)^2+\left(\sqrt{|a_2|}-a_2\right)^2+\ldots+\left(\sqrt{|a_n|}-a_n\right)^2$

Для проверки правильности решения задачи придумать тестовые примеры.

Замечание: При решении этой задачи всю последовательность чисел в памяти ПК хранить не надо.

Задание № 2

Дано вещественное число x. Вычислить:

$$\frac{(x-2)(x-4)(x-8)\ldots(x-64)}{(x-1)(x-3)(x-7)\ldots(x-63)}$$

Полученный результат проверить с помощью тестовых примеров. Тестовые примеры приведены в разделе *Рекомендации по выполнению* этого занятия.

Замечание: Как дополнительное задание можно учесть ограничения, накладываемые на значения переменной x.

Задание № 3

Вычислить длину кривой, соответствующую функции $f(x)=\sin(x)$ на отрезке $[a,b]$, приближенно заменив кривую ломаной, полученной в результате разбиения отрезка $[a,b]$ на n равных частей.

Полученный результат проверить с помощью тестовых примеров. Тестовые примеры приведены в разделе *Рекомендации по выполнению* этого занятия.

Задание № 4

Задано натуральное значение n, действительное значение x. Вычислить:

a) $\sum_{i=1}^{n} \frac{1}{i^2}$

b) $\prod_{i=1}^{n} \frac{i^2}{i^2+2i+3}$

c) $\sum_{i=1}^{n} (-1)^i \frac{x^{2i-1}}{i!}$

Полученный результат проверить с помощью тестовых примеров. Тестовые примеры приведены в разделе *Рекомендации по выполнению* этого занятия.

Задание № 5

Вычислить:

a) $\sum_{i=1}^{100} \sum_{j=1}^{50} \frac{1}{i+j^2}$

b) $\sum_{i=1}^{200} \sum_{j=1}^{i} \frac{1}{2j+i}$

c) $\sum_{i=1}^{50} \sum_{j=1}^{45} \sin(i^6+j^7)$

Полученный результат проверить с помощью тестовых примеров. Тестовые примеры приведены в разделе *Рекомендации по выполнению* этого занятия.

Задание № 6

Дано натуральное число *n*. Вычислить среднее арифметическое цифр, имеющихся в записи этого числа.

Задание № 7

Дано действительное число $x \neq 0$. Вычислить:

$$\cfrac{x}{x^2+\cfrac{2}{x^2+\cfrac{4}{x^2+\cfrac{8}{\cfrac{\ddots}{x^2+\cfrac{256}{x^2}}}}}}$$

Полученный результат проверить с помощью тестовых примеров. Тестовые примеры приведены в разделе *Рекомендации по выполнению* этого занятия.

Замечание: После того, как программа будет правильно вычислять значение дроби, добавить сохранение значения дроби в файл **`Result.dat`** при изменении значения переменной x в заданных пределах с заданным шагом. Пределы и шаг изменения значения переменной x вводятся с клавиатуры во время работы программы. Структура файла – результата: первый столбец — значение переменной x , второй столбец — вычисленное для данного значения переменной *x* значение цепной дроби. По полученному файлу построить график экспортировать его в какой-либо графический формат.

Задание № 8

Дано вещественное положительное число *b*. Последовательность $a_1, a_2, a_3, \ldots$ образована по закону:

$$a_1 = b\,, \qquad a_i = a_{i-1} - \frac{1}{\sqrt{i}}\,, \qquad i = 2, 3, \ldots .$$

Найти первый неотрицательный член последовательности.

Задание № 9

Последовательность задана по следующему правилу:

$$v_1 = v_2 = 0, \qquad v_3 = 1.5\,, \qquad v_i = \frac{i+1}{i^2+1} v_{i-1} - v_{i-2} v_{i-3}\,, \qquad i = 4, 5, \ldots$$

Дано натуральное n $(n \geqslant 4)$. Получить v_n.

Задание № 10

Записать реккурентную формулу и с ее помощью вычислить интеграл:

$$E_n = \int_0^1 x^n \mathrm{e}^{x-1} d\, x \qquad n = 1, 2, \ldots$$

При составлении реккурентной формулы учесть, что

$$E_n = x^n \mathrm{e}^{x-1}\Big|_0^1 - \int_0^1 n\, x^{n-1} \mathrm{e}^{x-1} d\, x = 1 - n\, E_{n-1}\,, \qquad n = 2, 3, \ldots \qquad E_1 = \frac{1}{\mathrm{e}}$$

Вычислить значение E_n, $n = 1, 2, \ldots, 20$. Результат вычислений вывести в виде таблицы значений на экран и в файл с именем **`Result.dat`**. Проверить правильность вычислений либо с помощью системы *Mathematica*, либо с помощью тестовых примеров.

Рекомендации по выполнению

Задание № 2

Тестовые примеры:

`x`	-3.6	-2.6	-1.6	-0.6	0.4	1.4	2.4	3.4
`res`	1.68827	1.83719	2.088	2.61989	4.76637	-3.23995	1.05325	-1.27956

Задание № 3

Длина дуги функции $\sin(x)$ на интервале $[0,3]$ приблизительно равна 3.62029.

Задание № 4

Тестовые примеры:

a). $n = 100$ $\quad res = 1.63498$

b). $n = 100$ $\quad res = 0.0000313524$

c). $n = 100$

`x`	-1.5	$-1.$	-0.5	0	0.5	1.	1.5
`res`	0.596401	0.632121	0.442398	0	-0.442398	-0.632121	-0.596401

Замечание: Формула общего члена заданного ряда достаточно простая $\left(a_i = (-1)^i x^{2i-1} / i\,!\right)$, но использовать ее не рационально, поскольку для каждого члена ряда нужно вычислить степень и факториал. Гораздо большей эффективности можно достичь, вычисляя член ряда с помощью реккурентного соотношения. Для нахождения этого соотношения найдем отношение:

$$\frac{a_i}{a_{i-1}}=\left((-1)^i\frac{x^{2i-1}}{i!}\right)\cdot\left(\frac{(-1)^{i-1}x^{2(i-1)-1}}{(i-1)!}\right)^{-1}=-\frac{(i-1)!\,x^{2i-2(i-1)}}{i!}=-\frac{x^2}{i}$$

Первый элемент этой последовательности, входящий в сумму, может быть вычислен очень просто: $a_1=-x$

Теперь становится понятен эффективный способ вычисления суммы n первых членов ряда.

Задание № 5

Тестовые примеры:

a). $res=24.6458$

b). $res=108.11$

c). $res=48.8664$

Задание № 6

Приведем возможный алгоритм решения задачи, записанный в виде псевдокода:

```
Вывести приглашения для ввода натурального числа n
Получить с клавиатуры значение для n
ЕСЛИ n > 0 ТО
  Присвоим начальные значения: сумма ← 0; количество ← 0
  ПОКА n /= 0 ВЫПОЛНИТЬ
    цифра ← n - (n / 10) * 10
     сумма ← сумма + цифра
     количество ← количество + 1
     n ← n / 10
  КОНЕЦ ПОКА
  среднее ← среднее / количество
  Вывод среднего значения на экран
ИНАЧЕ
  Вывод сообщения о недопустимом значении n
  Останов программы
КОНЕЦ ЕСЛИ
```

Задание № 7

Цепные дроби — это конструкции следующего вида:

$$R_4=\cfrac{a_1}{b_1+\cfrac{a_2}{b_2+\cfrac{a_3}{b_3+\cfrac{a_4}{b_4}}}}.$$

Обычно цепную дробь условно записывают в виде:

$$R_4=\frac{a_1}{b_1+}\,\frac{a_2}{b_2+}\,\frac{a_3}{b_3+}\,\frac{a_4}{b_4}.$$

Как правило, цепные дроби возникают при вычислении тех или иных высших трансцендентных функций. Если функция может быть представлена в виде цепной дроби, то этим обязательно надо воспользоваться, поскольку бесконечные цепные дроби обычно сходятся гораздо быстрее, чем ряды.

Разумеется, цепные дроби не следует вычислять в соответствии с явным выражением. Прямое вычисление цепной дроби не допускает рекурсии: если добавить к b_4 слагаемое a_5/b_5, то всю дробь придется пересчитывать заново. Правильная рекурсивная процедура такова:

$$R_n = \frac{A_n}{B_n}$$

$$A_n = b_n A_{n-1} + a_n A_{n-2}$$

$$B_n = b_n B_{n-1} + a_n B_{n-2}$$

Чтобы начать эту рекурсию, следует положить $A_0 = 0$, $B_0 = 1$, $A_1 = a_1$, $B_1 = b_1$. Проверить данную формулу можно с помощью метода математической индукции.

Тестовые примеры:

x	0.5	1.	1.5	2.	2.5	3.
res	1.04836	0.618718	0.528505	0.452539	0.382029	0.325643

Задание № 8

Покажем схематично, как можно провести вычисление.

```
1. a ← b
1. ДЛЯ i ОТ 2 ДО n ВЫПОЛНИТЬ
2.   a ← a - 1/sqrt(i)
     ! В правой части формулы значение переменной a - это
     ! предыдущий, «старый» элемент последовательности, а в
     ! левой части формулы - это «новый», следующий элемент
3. КОНЕЦ ДЛЯ
4. Вывод полученного элемента
```

Задание № 9

Покажем схематично, как можно провести вычисление.

```
1. Присвоить начальные значения первым членам последовательности
   v1 ← 0; v2 ← 0; v3 ← 1.5;
2. ДЛЯ i ОТ 4 ДО n ВЫПОЛНИТЬ
3.   Вычисление нового элемента последовательности по формуле
     vновое← (i+1)/(i^2+1)v3 - v2*v1
4.   «Обновление» переменных, учавствующих в вычислениях
     v1 ← v2; v2 ← v3; v3 ← vновое;
     ! После «обновления» в переменных v1, v2, v3 хранятся
     ! предыдущие элементы нужные для вычисления следующего
     ! элемента последовательности по той же формуле, которая
     ! записана в тексте программы
5. КОНЕЦ ДЛЯ
6. Вывод на экран полученного значения
```

Замечание: во время вычислений программа «помнит» не все элементы последовательности.

В памяти ПК находятся только те элементы последовательности, которые непосредственно нужны для вычисления последнего (три предпоследние элемента). При вычислении индексы элементов лучше рассматривать как указание на взаимное расположение элементов (новый, предыдущий, перед предыдущим и т.д.).

Задание № 10

Тестовые примеры:

n	1	2	3	4	5
E_n	0.367879	0.264241	0.207277	0.170893	0.145533

n	6	7	8	9	10
E_n	0.126802	0.112384	0.100932	0.0916123	0.0838771

n	11	12	13	14	15
E_n	0.0773522	0.0717733	0.0669477	0.0627322	0.0590175

n	16	17	18	19	20
E_n	0.0557193	0.0527711	0.0501199	0.0477228	0.0455449

Построение графиков таблично заданных функций

Для построения графиков функций можно использовать программу *GnuPlot*. Данная программа относится к фонду бесплатного программного обеспечения и реализована для разных операционных систем и типов компьютеров. Последние версии программы можно свободно получить из *Интернет* с сайта разработчика.

С программой обычно работают в однои из двух режимов:

- *интерактивном* — пользователь вводит команды построения и оформления графиков в командной строке программы (эти команды можно выбрать либо из соответствующих пунктов меню главного окна программы, либо с помощью панелей инструментов);
- *пакетном* — пользователь готовит специальный пакетный файл с командами, который передается как параметр программе в момент ее вызова. При этом программа работает в фоновом режиме, а построение, оформление и экспорт графика в нужный формат происходит автоматически.

Программа поставляется с хорошей справочной системой, где указана ВСЯ необходимая для работы информация. Кратко рассмотрим основы работы с программой.

После запуска программы в главном окне нужно указать рабочую папку и загрузить пакетный файл с командами. Для этого следует выбрать пункт *File | Change directory* главного окна программы или нажать кнопку *ChDir* панели инструментов. На экран будет выведено стандарное диалоговое окно выбора папки. После выбора нужной папки (в ней должны находиться файл с данными, по которым нужно построить график, и пакетный файл с командами *GnuPlot*) и подтвержения выбора в строке ввода программы будет введена и выполнена нужная команда перехода (см. **Рис. 3.1**).

После этого следует загрузить командный файл *GnuPlot* (текстовый файл с нужными командами). Обычно, командные файлы *GnuPlot* имеющий расширение имени файла **`PLT`**.

Для загрузки и выполнения нужного командного файла следует либо выбрать пункт меню *File | Open* главного окна программы, либо нажать кнопку *Open* панели задач. На экран будет выведено стандартное окно выбора файла, в котором нужно указать требуемый пакетный файл. После подтвержения выбора программа *GnuPlot* выполнит все команды, находящиеся в файле. При этом в строке ввода главного окна программы будет указана выполненная команда загрузки пакетного файла.

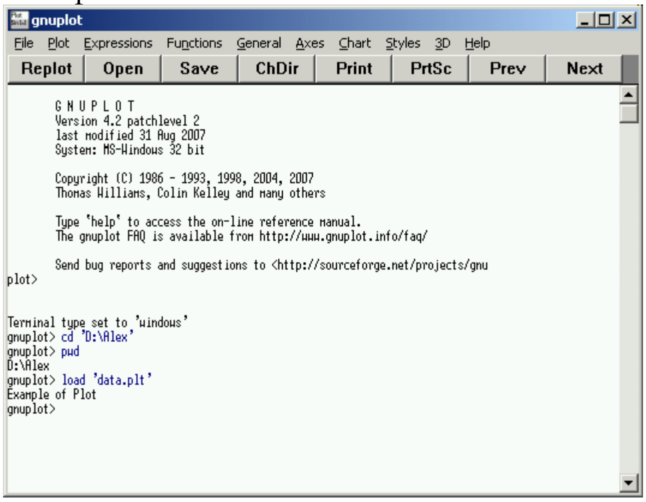

Рисунок 3.1. Главное окно программы *GnuPlot*

Пример пакетного файла для построения двумерного графика по набору точек, хранящихся в файле приведен ниже. В пакетном файле указаны краткие комментарии к основным командам. При необходимости все о командах можно узнать в справочной системе программы.

```
print "Example of Plot" # вывод текста в окно редактора
reset
set nogrid  # без сетки на графике
set nokey   # без автоматической легенды
set xrange [0.5:5.5]  # диапазон для x на графике
set yrange [0.0:1.7]  # диапазон для y на графике
set tics out          #  штрихи у осей наружу
set xtics border nomirror norotate 0.0, 1, 5.5  # цифры у оси x
set ytics border nomirror norotate 0.0, 0.5, 1.6 # цифры у оси y
# количество разбиений большого интервала по оси x
set mxtics 10
# количество разбиений большого интервала по оси y
set mytics 5
# текстова метка на графике
# (это одна строка файла. ↵ - символ продолжения строки)
set label 1 "Chain Fraction" at 0.7, 1.55 left norotate font ↵
"Arial Italic,24"
# построение графика в окне
plot 'Data1.dat' using 1:2 with lines linetype 1 linewidth 3
pause -1 "Press OK to make eps file or Exit to stop..."
# экспорт графика в emf формат
set terminal emf color solid linewidth 5 "Times Italic" 24
set output 'Fig1.emf'
replot
# восстановление выводного устройства
set terminal windows
set output
pause -1 "Press any key to continue..."
reset
```

Тема № 4. Программы комбинированной структуры

Это занятие посвящено программам, включающим циклы и разветвления. На занятии необходимо научиться численно вычислять пределы, суммы и цепные дроби с заданной точностью, а также практически познакомиться с особенностями целочисленных вычислений.

Список заданий

Задание № 1

Дано натуральное число n , действительное число *x*. Среди чисел $\exp\left(x^{-2k}\right)\sin\left(x^{3k}\right)$, $(k=1,\ldots,n)$ найти ближайшее к какому–нибудь целому.

Задание № 2

Вычислить бесконечную сумму с заданной точностью ε $(\varepsilon>0)$. Считать, что требуемая точность достигнута, если вычислена сумма некоторых первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε — это и все последующие слагаемые можно не учитывать. Вычислить:

a) $\sum_{i=1}^{\infty}\frac{1}{i(i+1)}$

b) $\sum_{i=0}^{\infty}\frac{(-1)^i}{i!}$

c) $\sum_{i=1}^{\infty}\frac{(-1)^{i+1}}{i(i+1)(i+2)}$

Задание № 3

Числа Фибоначчи получаются с помощью следующего соотношения:

$$f_1=f_2=1,\qquad f_n=f_{n-1}+f_{n-2},\qquad n\geqslant 3\,.$$

С заданной погрешнстью ε численно вычислить предел выражения:

$$\lim_{n\to\infty}\left(\frac{f_n}{f_{n-1}}\right)$$

Сравнить полненный ответ с *золотым сечением*:

$$\frac{\sqrt{5}+1}{2}.$$

Задание № 4

Написать программу, которая вычисляет наибольший общий делитель двух целых чисел. Наибольший общий делитель $(НОД)$ неотрицательных целых чисел может быть вычислен при помощи алгоритма Евклида. Пусть m и n — одновременно не равные нулю целые неотрицательные числа и пусть $m\geqslant n$. Тогда, если $n=0$ то $НОД(n,m)=m$, а если $n\neq 0$, , то для чисел m, n и r, где r — остаток от деления m на n,

выполняется равенство $НОД(m,n)=НОД(n,r)$.
Например, $НОД(15{,}6)=НОД(6,3)=НОД(3,0)=3$.

Задание № 5

Написать программу, которая вычисляет число π с заданной пользователем точностью. Для этого воспользуйтесь тем, что значение частичной суммы ряда

$$1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}\ldots$$

при суммировании достаточно большого количества слагаемых приближается к $\pi/4$.
Замечание: Сравнить полученное значение, со значением, возвращаемым стандартными функциями $(\pi=\mathrm{ACOS}(-1.0))$. Для удобства работы, при проверке расчетов использовать небольшие значения точности.

Задание № 6

Число e – основание натуральных логарифмов (напомним, что оно приблизительно равно $2.718281828)$, может быть получено в результате предельного перехода по формуле:

$$\lim_{x\to 0}(1-x)^{-1/x}$$

Напишите программу, с заданной точностью демонстрирующую справедливость этого утверждения.

Задание № 7

Написать программу, которая проверяет, является ли число, введеное пользователем, простым. Натуральное число называется простым, если его делители только 1 и само это число.

Задание № 8

Для произвольного (положительного и отрицательного, большого и малого) заданного аргумента x, напишите программу вычисления функций $\sin(x)$ и $\cos(x)$ посредством вычисления бесконечных сумм с заданной точностью ε $(\varepsilon>0)$.

$$\sin(x)=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}+\ldots$$

$$\cos(x)=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\ldots$$

Считать, что требуемая точность достигнута, если вычислена сумма нескольких первых слагаемых и очередное слагаемое оказалось по модулю меньше, чем ε, — это и все последующие слагаемые можно уже не учитывать. После вычисления функций проверить точность, с которой выполняется тождество $\sin(x)^2+\cos(x)^2=1$.

Задание № 9

Для заданного значения $x>1$ вычислить y — значение корня квадратного от x по итерационной формуле:

$$y_i=\frac{1}{2}\left(y_{i-1}+\frac{x}{y_{i-1}}\right)$$

с заданной погрешностью ε, задав начальное приближение $y_0 = x$. Сравнить с результатом использования встроенной функции. Сколько итераций пришлось выполнить?

Задание № 10

Найти 100 первых простых чисел. Натуральное число называется простым, если его делители только 1 и само это число.

Задание № 11

Найти положение минимума функции комплексного аргумента, заданной разложением в ряд:

$$J_0(z) = \sum_{k=0}^{\infty} \frac{(-1)^k (z^2/4)^k}{(k!)^2}$$

в области $0 \leqslant \mathrm{Re}(z) \leqslant 3$, $-1 \leqslant \mathrm{Re}(z) \leqslant 1$, вычисляя таблицу значений с заданным шагом h.

Задание № 12

С заданной точностью вычислить предел выражения:

$$\lim_{x \to 0} \left(\frac{1 + \mathrm{tg}\ x}{1 + \sin\ x} \right)^{(\sin\ x)^{-2}}$$

Задание № 13

Натуральное число называется совершенным, если оно равно сумме всех своих делителей, за исключением себя самого, но включая 1. Так, число 6 — совершенное, так как $6 = 1 + 2 + 3$, а число 8 — не совершенное, т.к. $8 \neq 1 + 2 + 4$. Найти все совершенные числа, меньшие заданного натурального числа n.

Задание № 14

Для заданного x с заданной точностью ε вычислить

$$\arctan\ x\ =\ \frac{x}{1+}\frac{x^2}{3+}\frac{4x^2}{5+}\frac{9x^2}{7+}\ldots$$

Результат вычислений сравнить со значением, полученным встроенной функцией (`ATAN(x)`). Трактуя это как «точное» значение получить абсолютную и относительную ошибки. Сравнить с заданной точностью расчетов ε.

Рекомендации по выполнению

Задание № 1

Для $x = 2.5$, $n = 16$ ближайшим к целому будет число 0.998063 при $k = 6$.

Задание № 2

Тестовые примеры:

a). 1.0 (точное значение – 1);

b). 0.367879 (точное значение – $1/e$);

c). 0.136294 (точное значение – $(8 \cdot \ln(2) - 5)/4$);

Задание № 5

Расчетную формулу можно записать в виде: $\sum_{n=0}^{\infty} \frac{(-1)^n}{2 \cdot n+1}$.

Задание № 8

Расчетные формулы можно записать в виде:

$$\sin(x) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2\mathrm{n}-1}}{(2\mathrm{n}-1)!}, \qquad \cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2\mathrm{n}}}{(2\mathrm{n})!}.$$

Задание № 11

Примерное положение минимума в заданной области: $\mathrm{Re}(z_{min}) = 2.40483$, $\mathrm{Im}(z_{min}) = 0$.

Задание № 12

Тестовый результат:1

Задание № 14

Для расчетов можно использовать следующие формулы: $a_1 = x$, $a_n = (n-1)^2 x^2$ для $n > 1$; $b_n = 2n - 1$.

Тема № 5. Функции и подпрограммы. Основы работы

На данном занятии необходимо познакомиться с основными программными единицами *Фортрана*: функциями и подпрограммами. На занятии необходимо научиться разделять задачу на программные единицы и обмениваться данными между ними.

Список заданий

Задание № 1

Дано действительное число y. Получить значение выражения:

$$\frac{1.7\cdot f(0.25)+2\cdot f(1+y)}{6-f(y^2-1)}$$

где функция $f(x)$ определена так:

$$f(x)=\frac{\sum_{k=0}^{16}\frac{x^{2k+1}}{(2k+1)!}}{\sum_{k=0}^{10}\frac{x^{2k}}{(2k)!}}$$

Тестовое значение для проверки работы программы указано в *Рекомендациях по выполнению* этого задания.

Задание № 2

Написать подпрограмму решения квадратного уравнения. В качестве параметров подпрограмма принимает коэффициенты квадратного уравнения, а возвращает решения квадратного уравнения (вещественные или комплексные) и переменную, указывающую на количество корней и тип решения (корни могут быть вещественные или комплексные; кроме того, уравнение может не иметь решения).

Замечание: возможый вариант заголовка процедуры и тестовые значения для проверки программы приведены в разделе *Рекомендации по выполнению* этого задания.

Задание № 3

Даны натуральное число n, действительные числа x_1, y_1, x_2, y_2, …, x_n, y_n. Найдите площадь n –угольника, вершины которого при некотором последовательном обходе имеют координаты (x_1, y_1), (x_2, y_2), …, (x_n, y_n) (см. **Рис. 5.1**).

Замечание: Для решения задания определите процедуру вычисления площади треугольника по координатам его вершин.

Задание №4

Даны действительные числа x_1, y_1, x_2, y_2, …, x_6, y_6. Точки с координатами (x_1, y_1), (x_2, y_2), (x_3, y_3) рассматриваются как вершины первого треугольника, точки с координатами (x_4, y_4), (x_5, y_5), (x_6, y_6) — второго треугольника. Выяснить, верно ли, что первый треугольник целиком содержится во втором, и если да, определить площадь области, принадлежащей внешнему треугольнику и не принадлежащей внутреннему. (Заштрихованная область на **Рис. 5.2**.)

Замечание: Возможый набор процедур, необходимый для решения задачи приведен в разделе *Рекомендации по выполнению* этого задания.

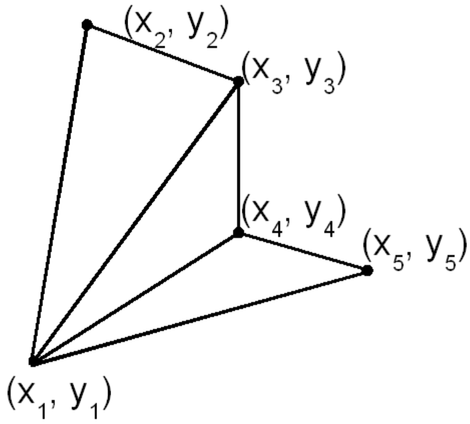

Рисунок 5.1. Вычисление площади многоугольника при помощи разбиения его на треугольники.

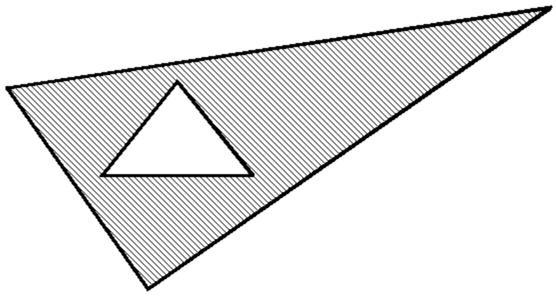

Рисунок 5.2. Область, площадь которой требуется найти в **Задании №4**

Задание № 5

Напишите функцию, предназначенную для определения наибольшего общего делителя $NOD(x, y)$, двух натуральных чисел x и y с помощью алгоритма Евклида: $NOD(A, B)=NOD(B\,mod\,A, A)$, если $A\neq 0$; $NOD(0,B)=B$.

Замечание: Написать нерекурсивную и рекурсивную процедуры вычисления наибольшего общего делителя. Придумать тестовые примеры для проверки правильности решения задачи.

Задание № 6

Для заданного вещественного числа $a>0$ вычислить выражение:

$$\frac{\sqrt[3]{a}-\sqrt[6]{a^2+1}}{1+\sqrt[7]{3+a}}$$

Корни $y=\sqrt[k]{a}$ вычислить с точностью $\varepsilon=0.00001$ по следующей итерационной формуле:

$$y_{n+1}=y_n+\frac{1}{k}\left(\frac{x}{y_n^{k-1}}-y_n\right)$$

$y_0=1$, $n=0,1,2,\ldots,$ приняв за ответ приближение y_{n+1}, для которого $|y_{n-1}-y_n|<\varepsilon$.

Замечание: Для проверки правильности выполнения программы вычислите данное выражение с помощью встроенных функций *Фортрана*.

Задание № 7

Для вещественного числа t вычислите с точностью ε величину:

$$\sqrt[4]{1-\frac{\cos^4 t}{4}}+\sqrt[5]{1+\frac{\sin t}{2}}\cdot\sqrt[9]{1+\left(\frac{\ln t}{\exp t}\right)^2}$$

Для вычисления корней используйте следующий ряд Тейлора:

$$(1+x)^a=1+\frac{a}{1!}x+\frac{a(a-1)}{2!}x^2+\frac{a(a-1)(a-2)}{3!}x^3+\ldots$$

$$|x|\leqslant 1,\qquad a>0.$$

Замечание: Для проверки правильности выполнения программы вычислите данное выражение с помощью встроенных функций *Фортрана*.

Задание № 8

Написать подпрограммы вычисления функций Эйри комплексного аргумента, используя разложения в степенной ряд:

$$Ai(z)=c_1 f(z)-c_2 g(z)$$

$$Bi(z)=\sqrt{3}\left(c_1 f(z)+c_2 g(z)\right)$$

$$f(z)\;=\;1+\frac{1}{3!}z^3+\frac{1\cdot 4}{6!}z^6+\frac{1\cdot 4\cdot 7}{9!}z^9+\ldots\;=\;\sum_0^\infty 3^k\left(\frac{1}{3}\right)_k\frac{z^{3k}}{(3k)!}$$

$$g(z)\;=\;z+\frac{2}{4!}z^4+\frac{2\cdot 5}{7!}z^7+\frac{2\cdot 5\cdot 8}{10!}z^{10}+\ldots\;=\;\sum_0^\infty 3^k\left(\frac{2}{3}\right)_k\frac{z^{3k+1}}{(3k+1)!}$$

здесь
$c_1\approx 0.355028053887817$, $c_2\approx 0.258819403792807$, а $(z)_k$ обозначает символ Похгаммера: $(z)_0=1$; $(z)_n=z\cdot(z+1)\cdot(z+2)\cdot\ldots\cdot(z+n-1)$.
Тестовые значения для проверки работы программы приведены в *Рекомендациях по выполнению* этого задания.

Задание № 9

Напишите программу, определяющую примерное месторасположение комплексного корня с минимальными положительными вещественными и мнимыми частями для уравнения:

$$f(z)=0$$

где $f(z)=\ln(\mathrm{Re}(z)+i\cdot(\mathrm{Im}(z)-3))-5\cdot(\mathrm{Re}(z)+i\cdot(\mathrm{Im}(z)-4))+\sinh(\mathrm{Re}(z)+i\cdot(\mathrm{Im}(z)-5))$
Вычислить минимальное значение $|f(z)|$ в найденной окрестности корня.
Поиск примерного месторасположения корня и его примерное значение вычисляются «*табличным* (*графическим*)» методом. Значение элементарных числовых функций вычислить с заданной точностью специально написанными процедурами, основанными на разложениях:

$$\ln(z)=2\left(\frac{z-1}{z+1}+\frac{1}{3}\left(\frac{z-1}{z+1}\right)^3+\ldots+\frac{1}{2n-1}\left(\frac{z-1}{z+1}\right)^{(2n-1)}+\ldots\right)\qquad \mathrm{Re}(z)\geqslant 0\qquad z\neq 0$$

$$\sinh(z)=z+\frac{z^3}{3!}+\frac{z^5}{5!}+\ldots+\frac{z^{2n+1}}{(2n+1)!}+\ldots\qquad |z|<\infty$$

Примерное расположение корня, структура файла данных и скрипт программы *GnuPlot*, предназначенный для построния двумерного графика приведен в *Рекомендациях по выполнению* этого задания.

Задание № 10

Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int \frac{d\,x}{\sin^n x} = \begin{cases} -\frac{1}{n-1}\frac{\cos x}{\sin^{n-1} x} + \frac{n-2}{n-1}\int \frac{d\,x}{\sin^{n-2} x}, & n \geqslant 2 \\ \ln tg\frac{x}{2}, & n=1 \\ x, & n=0 \end{cases}$$

Тестовые значения для проверки работы программы приведены в *Рекомендациях по выполнению* этого задания.

Рекомендации по выполнению

Задание № 1

Тестовое значение для проверки работы программы $f(0.5)\approx 0.462117$, $\text{result} \approx 0.335585$.

Задание № 2

Для выбора варианта решения из множества возможных рекомендуется воспользоваться оператором `CASE`.

Возможный вариант заголовка подпрограммы:

```
subroutine SolveQuadraticEquation(a, b, c, x1, x2, par)
```

здесь:

- `a, b, c` – коэффициенты квадратного уравнения $a\cdot x^2 + b\cdot x + c = 0$. Входные параметры;
- `x1, x2` – корни квадратного уравнения в случае вещественных корней и вещественная и мнимая части решения в случае комплексных корней, соответственно. Выходные пареметры;
- `par` – параметр, определяющий тип и количество решений. Выходной параметр.
  - ✔ `par = 0`, если нет решений (набор коэффициентов не определяет уравнение)
  - ✔ `par = 1`, если одно вещественное решение (уравнение линейно)
  - ✔ `par = 2`, если два вещественные решения
  - ✔ `par = -2`, если комплексные решения

Тестовые значения для проверки правильности работы программы:

a	b	c	x_1	x_2
0	0	0,2	Нет решения	
0	1	2	Одно решение, -0.5	
1	2	0	-2	0

1	5	-3	-5.54138	0.541381
1	2	3	$-1.0-1.41421\cdot i$	$-1.0+1.41421\cdot i$

Задание № 4

Для решения задачи определить процедуру, позволяющую выяснить, лежат ли две точки в одной полуплоскости относительно прямой (**Тема № 2**, **Задание № 8**), процедуру вычисления расстояния между двумя точками, а также процедуру вычисления площади треугольника по трем сторонам.

Задание № 8

Для решения этого задания необходимо аккуратно получить расчетные формулы «на бумаге». Тестовое значение для проверки правильности работы программы:

$z=0.1+i\cdot 0.3 \quad Ai(z)\approx 0.327548-i\cdot 0.0785021, \quad Bi(z)\approx 0.657196+i\cdot 0.132286.$

Задание № 9

Данную задачу рекомендуется решать по этапам. Сначала нужно написать процедуры для расчета элементарных функций с заданной точностью для вещественных аргументов и проерить их с помощью встроенных функций *Фортрана*. Затем следует переделать их для работы с вещественными аргументами. Далее, написать и проверить работу процедуры вычисления $f(z)$ для заданного аргемента, а лишь затем формировать файл данных, строить график и находить табличным методом минимум.

Примерное расположение корня: $x\approx 3.55427 \quad y\approx 5.36438$

Стуктура файла данных, для программы *GnuPlot*. Для построения трехмерного графика по набору точек файл с данніми должен содержать три столбца. Первый столбец – значения вещественной части z, второй столбец – значения мнимой части z, третий столбец – значение $|f(z)|$. Различные значения в первом столбце отделяются пустой строкой. **Например:**

```
0.9200000E+01   -.1000000E+00   0.2383385E+00
0.9200000E+01   0.0000000E+00   0.8444091E-02
0.9200000E+01   0.1000000E+00   0.2383385E+00

0.9210000E+01   -.1000000E+00   0.2390888E+00
0.9210000E+01   0.0000000E+00   0.8132014E-02
0.9210000E+01   0.1000000E+00   0.2390888E+00

0.9220000E+01   -.1000000E+00   0.2398385E+00
0.9220000E+01   0.0000000E+00   0.7820232E-02
0.9220000E+01   0.1000000E+00   0.2398385E+00
```

и т.д.

Возможный вид скрипта программы *GnuPlot*.

```
reset
set samples 20
set isosamples 21
set view 60,30,1,1
set xlabel "Re(z)" -5,-2
set ylabel "Im(z)" 4,-1
```

```
set zlabel "Abs(F)"
set title "3D Function Profile"
set contour
set cntrparam bspline
#set cntrparam levels 11
splot 'Profile3D.dat' u 1:2:3  w l 1
pause -1  "Hit ENTER to continue"

set grid
set view 0,90,1,1
set xlabel "Im(z)" -5,-2
set ylabel "Re(z)" 4,-1
set title "Contour Function Profile"
set nosurface
set cntrparam levels auto 25
splot 'Profile3D.dat' u 2:1:3  w l 1
pause -1  "Hit ENTER to end"
```

Задание № 10

Тестовые значения для проверки правильности работы программы:

- $a=0.1$; $b=0.6$; $n=0$; result$=0.5$
- $a=0.1$; $b=0.6$; $n=1$; result$=1.82157$
- $a=0.1$; $b=0.6$; $n=2$; result$=8.50495$
- $a=0.1$; $b=0.6$; $n=3$; result$=49.5328$
- $a=0.1$; $b=0.6$; $n=4$; result$=337.473$

Тема № 6. Многофайловые программы

На данном занятии необходимо познакомиться с основными численными методами, применяемыми как самостоятельно, так и в составе более сложных численных алгоритмов: методами вычисления производных, интегралов и решения нелинейных уравнений. Кроме того, нужно научиться работать с проектами *Фортрана*, состоящими из нескольких файлов.

Список заданий

Задание № 1

Написать процедуру, которая вычисляет производную заданной пользователем функции $f(x)$ заданным методом (по определению, с помощью 3-х и 5-и точечной схемы) с заданной точностью ε.

Замечание: Для расчетов взять функцию $f(x)=x^2-\mathrm{e}^x$. Сравнить полученное значение с точным (аналитически вычисленным) значением производной в заданной точке. Основные расчетные формулы и рекомендации по организации программы приведены в *Рекомендациях по выполнению* этого задания. В качестве дополнительного задания вычислите вторую производную от заданной функции и сравните с аналитически вычисленным значением.

Задание № 2

Написать процедуру, которая вычисляет определенный интеграл

$$I=\int_a^b f(x)\ dx$$

заданным методом (левых прямоугольников, правых прямоугольников и средних прямоугольников) с заданной точностью ε. Для расчетов взять функцию $f(x)=\sin(2\sqrt{(x)})$. Сравнить результат с аналитически вычисленным значением.

Замечание: Значение неопределенного интеграла от заданной функции (с точностью до константы интегрирования) равно:

$$\frac{1}{2}\sin(2\sqrt{x})-\sqrt{x}\cos(2\sqrt{x})$$

Основные расчетные формулы приведены в *Рекомендациях по выполнению* этого задания.

Задание № 3

Вычислить интеграл с заданной точностью составным методом трапеций

$$\int_{0.4}^{1.2}\frac{dx}{\sqrt{3+x^2}}$$

Полученный ответ сравните с тестовым значением: 0.41784

Замечание: Основные расчетные формулы приведены в *Рекомендациях по выполнению* этого задания.

Задание № 4

Вычислить интеграл с заданной точностью составным методом Симпсона

$$\int_{1.3}^{2.1} \frac{\sin(x^2-1)}{2\sqrt{x}}\, dx$$

Полученный ответ сравните с тестовым значением: 0.213079

Замечание: Основные расчетные формулы приведены в *Рекомендациях по выполнению* этого задания.

Задание № 5

С помощью ранее разработанных процедур вычислить двумерный интеграл с заданной точностью:

$$\int_0^1 \int_0^1 \frac{x^2}{1+y^2}\, dx\, dy$$

Полученный ответ сравните с тестовым значением: $\frac{\pi}{12}$

Задание № 6

С помощью ранее разработанных процедур вычислить с заданной точностью интеграл на полубесконечном интеграле:

$$\int_0^{\infty} e^{-x^2} + \frac{e^{-(x-3)^2}}{x+2}\, dx$$

Полученный ответ сравните с тестовым значением: 1.24827

Замечание: Методика вычислений приведена в разделе *Рекомендациях по выполнению* этого задания.

Задание № 7

Написать процедуру, которая находит решение нелинейного уравнения методом деления отрезка пополам. С помощью этой процедуры с заданной точностью найти первый положительный корень уравнения

$$x^2 \cos(2x) = -1$$

Полученный ответ сравните с тестовым значением.

Замечание: Основные расчетные формулы и тестовое значение приведены в *Рекомендациях по выполнению* этого задания.

Задание № 8

Написать процедуру, которая находит решение нелинейного уравнения методом касательных. С помощью этой процедуры с заданной точностью найти решение уравнения

$$3x - \cos(x) - 1 = 0$$

Полученный ответ сравните с тестовым значением.

Замечание: Основные расчетные формулы и тестовые примеры приведены в *Рекомендациях* *по выполнению* этого задания.

Задание № 9

Написать процедуру, которая находит решение нелинейного уравнения методом секущих. С помощью этой процедуры с заданной точностью найти первый положительный корень уравнения

$$\operatorname{tg}(0.3x+0.4)=x^2$$

Полученный ответ сравните с тестовым значением.

Замечание: Основные расчетные формулы и тестовые примеры приведены в *Рекомендациях по выполнению* этого задания.

Задание № 10

Написать процедуру, которая находит решение нелинейного уравнения методом обратной квадратичной интерполяции. С помощью этой подпрограммы с заданной точностью найти первые два корня уравнения

$$x^3-e^x=0$$

Полученный ответ сравните с тестовыми значениями.

Замечание: Основные расчетные формулы и тестовые примеры приведены в *Рекомендациях по выполнению* этого задания.

Задание № 11

Написать процедуру, которая находит решение нелинейного уравнения методом Мюллера. С помощью этой процедуры с заданной точностью найти комплексный корень с минимальными положительными вещественной и мнимой частями для уравнения

$$e^z-0.2z+1=0$$

Полученный ответ сравните с тестовым значением **.**

Замечание: Основные расчетные формулы и тестовые примеры приведены в *Рекомендациях по выполнению* этого задания.

Задание № 12

При помощи разработанных процедур с заданной точностью решить уравнение:

$$K(x)-p_1=0$$

где:

$$K(x)=\int_0^{\pi/2}\frac{1}{\sqrt{1-x\sin^2 t}}\,dt$$

здесь параметр p_1 может принимать значения из диапазона $p_1 \geqslant \pi/2$. Для заданного значения параметра сначала локализуйте корень уравнения. Полученный ответ сравните с тестовым значением.

Замечание: Основные расчетные формулы и тестовые примеры приведены в *Рекомендациях по выполнению* этого задания.

Рекомендации по выполнению

Задание № 1

Для того, чтобы численно с заданной точностью ε получить значение производной функции $f(x)$ необходимо создать последовательность:

$$f'(x) \approx D_n,$$

где D_n определяется методом, который используется для вычислений:

- по определению:

$$D_n = \frac{f\left(x + 10^{-k} h\right) - f(x)}{10^{-k} h}$$

- 3-х точечная схема:

$$D_n = \frac{f\left(x + 10^{-k} h\right) - f\left(x - 10^{-k} h\right)}{2 \cdot 10^{-k} h}$$

- 5-и точечная схема:

$$D_n = \frac{-f\left(x + 2 \cdot 10^{-k} h\right) + 8 \cdot f\left(x + 10^{-k} h\right) - 8 \cdot f\left(x - 10^{-k} h\right) + f\left(x - 2 \cdot 10^{-k} h\right)}{12 \cdot 10^{-k} h}$$

для $k = 0, 1, 2, \ldots, n$ до тех пор, пока не будет выполнено хотя бы одно неравенство:

$$|D_{n+1} - D_n| \geqslant |D_n - D_{n-1}|, \quad \text{или} \quad |D_n - D_{n-1}| < \varepsilon$$

При этом производная функции f в указанной точке x будет приближенно равна D_n.
Рекомендуется попробовать реализовать вычислительную программу, состоящую из следующих процедур:

- функция, производную которой нужно вычислить: `function fun(x)`
- функция, вычисляющая производную с заданным шагом (для 3-х точечной схемы): `function der3pt(x, h, f)`, где `x` – точка, в которой вычисляется производная, `h` – шаг, с которым нужно провести вычисление, `f` – функция, производная которой вычисляется программой
- функция, вычисляющая производную в заданной точке с заданной точностью от заданной функции заданным методом: `function D(x, tol, f, meth)`, где `x` – точка, в которой вычисляется производная, `tol` – точность, с которой нужно провести вычисление, `f` – функция, производная которой вычисляется программой, `meth` – метод с заданным шагом, с помощью которого производится вычисление
- основная программа, организующая вычисление

Преимущества такой организации программы:

- Легко можно вычислить производную любой функции — она передется как параметр и главное, чтобы она принимала один аргумент — значение вещественного типа и возвращала вещественный результат.
- Можно очень просто управлять методом, с помощью которого происходит вычисление — метод просто передается как параметр при вызове функции дифференцирования.
- Очень просто расширять программу — можно дописать новый метод вычисления производной с заданным шагом и передать как аргумент при вызове функции диференцирования. Главное — согласовать интерфейс вызова.

Задание № 2

Для вычисления определеного интеграла

$$\int_a^b f(x)\ dx$$

можно воспользоваться составным методом прямоугольников. Пусть интервал интегрирования $[a,b]$ разбит на n частей с шагом $h=\frac{b-a}{n}$. Тогда приближенное значение интеграла можно получить:

- методом левых прямоугольников:

$$\int_a^b f(x)\ dx \quad \approx \quad h\cdot\sum_{i=0}^{n-1} f(x_i)$$

- методом правых прямоугольников:

$$\int_a^b f(x)\ dx \quad \approx \quad h\cdot\sum_{i=1}^{n} f(x_i)$$

- методом средних прямоугольников:

$$\int_a^b f(x)\ dx \quad \approx \quad h\cdot\sum_{i=0}^{n-1} f\left(x_i+\frac{h}{2}\right)$$

Здесь $x_i=a+i\cdot h \quad (i=0,1,\dots,n)$. Будем считать, что интеграл вычислен с заданной точностью ε, если при последовательном удвоении шагов будет выполнено условие $|I_n-I_{2n}|<\varepsilon$. Здесь I_n и I_{2n} — интегральные суммы, посчитанные для n и $2n$ шагов разбиения.

Задание № 3

Для вычисления определеного интеграла

$$\int_a^b f(x)\ dx$$

можно воспользоваться составным методом трапеций. Пусть интервал интегрирования $[a,b]$ разбит на n частей с шагом $h=\frac{b-a}{n}$. Тогда приближенное значение интеграла можно получить по формуле:

$$\int_a^b f(x)\ dx \quad \approx \quad h\cdot\left(\frac{1}{2}y_0+y_1+y_2+\ldots+y_{n-1}+\frac{1}{2}y_n\right)$$

Здесь $x_i=a+i\cdot h \quad (i=0,1,\dots,n)$, $\quad y_i=f(x_i)$ — значения функции на концах отрезков. Будем считать, что интеграл вычислен с заданной точностью ε, если при последовательном удвоении шагов будет выполнено условие $|I_n-I_{2n}|<\varepsilon$. Здесь I_n и I_{2n} — интегральные суммы, посчитанные для n и $2n$ шагов разбиения.

Задание № 4

Для вычисления определеного интеграла

$$\int_a^b f(x)\ dx$$

можно воспользоваться составным методом Симпсона. Пусть интервал интегрирования $[a,b]$ разбит на n частей с шагом $h=\frac{b-a}{n}$. Тогда приближенное значение интеграла можно получить по формуле:

$$\int_a^b f(x)\ dx \quad \approx \quad \frac{h}{6}\cdot\left(y_0+y_n+4\cdot\left(y_{1/2}+y_{3/2}+\ldots+y_{n-1/2}\right)+2\cdot\left(y_1+y_2+\ldots+y_{n-1}\right)\right)$$

Здесь $x_i=a+i\cdot h \quad (i=0,1,\ldots,n)$, $\quad y_i=f\left(x_i\right)$ — значения функции на концах отрезков. Будем считать, что интеграл вычислен с заданной точностью ε, если при последовательном удвоении шагов будет выполнено условие $|I_n-I_{2n}|<\varepsilon$. Здесь I_n и I_{2n} — интегральные суммы, посчитанные для n и $2n$ шагов разбиения.

Задание № 6

Вычисление интеграла при бесконечных пределах зависит от поведения подынтегральной функции на бесконечности. Если подынтегральная функция убывает достаточно быстро, как в приведенном задании, то легко численно вычислить верхний предел интегрирования, дальше которого идти не имеет смысла. В это случае интегрирование производится как-бы по конечному интервалу. Затем можно увеличить выбранный верхний предел интегрирования в полтора – два раза и убедиться, что изменение ответа не превышает заказанную точность. Если это не так, то первоначальная оценка верхнего предела неверна, и следует повторить проделанную вычислительную процедуру, т.е. снова увеличить верхний предел интегрирования, и так до тех пор, пока не будет достигнута требуемая точность.

Общие замечания по методам решения нелинейных уравнений

Поиск корней (в общем случае комплексных) нелинейного уравнения с одним неизвестным

$$f(x)=0$$

выполняется в два этапа. На первом этапе осуществляется *локализация корня*, т.е. определяется такой отрезок $[a,b]$, на котором исследуемая функция $f(x)$ имеет строго один корень. Этот отрезок называют *отрезком локализации* корня $\bar{x}$. На втором этапе выполняется поиск корня на отрезке локализации. Локализацию корня будем проводить графическим методом.

При вычислении корня нелинейного уравнения оценку точности нахождения корня можно проводить нексолькими способами. Если уравнение «хорошее», то эти методы практически эквивалентны. Обычно, корень уравнения считается найденным, если либо выполняется неравенство $|x^{(k+1)}-x^{(k)}|<\varepsilon$, где $x^{(k+1)}, x^{(k)}$ – два последовательных приближения к корню (или границы интервала локализации корня), ε – требуемая точность расчетов, либо выполняется неравенство $|f(x^{(k)})|<\varepsilon$. Иногда при расчетах используют комбинированный критерий.

Задание № 7

Тестовое значение: 1.18321

Для применения метода дихотомии необходимо корректное указание $[a,b]$ – отрезка локализации корня. Предполагается, что $f(x)$ непрерывна в окрестности корня и имеет различные знаки на концах интервала локализации $\left(f(a)\cdot f(a)<0\right)$. На каждой итерации отрезок локализации корня делится пополам и в качестве нового отрезка выбирается та из половин, на концах которой функция $f(x)$ имеет значения разных знаков. Обычно, вычисления прекращают когда длинна нового отрезка локализации становится меньше требуемой точности расчетов $|a-b|<\varepsilon$. Тогда в качестве корня можно использовать значение середины финального отрезка локализации корня. Метод дихотомии обладает глобальной сходимостью, что значит, если на заданном интервале функция непрерывна и имеет разные знаки на концах, то корень будет обязательно найден.

Рекомендации: во время вычислений центр отрезка локализации корня лучше вычислять по формуле

$$x=a+\frac{b-a}{2}$$

Задание № 8

Тестовое значение: 0.607102

Метод касательных (его еще называют методом Ньютона) — это метод, использующий специальную линеаризацию задачи, сводящую решение исходного нелинейного уравнения к решению последовательности линейных уравнений. В данном методе для новое приближение к корню уранения $f(x)=0$ вычисляется по формуле:

$$x^{(k+1)}=x^{(k)}-\frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}.$$

Следует учесть, что данный метод не всегла приводит к успеху. Так, например, если производная мала, то метод может не сойтись.

Рекомендации: во время написания программы учесть, что производная, необходимая для вычисления нового приближения к корню, может быть вычислена или аналитически, или численно. Это можно предусмотреть с помощью необязательного формального параметра – функции, которая аналитически вычисляет производную. Если этот аргумент указан при вызове процедуры, то производная вычисляется аналитически, если нет — то вычисляется численно с помощью ранее разработанных процедур вычисления производной.

Задание № 9

Тестовое значение: 0.886345

Метод секущих можно рассматривать как модификацию метода Ньютона, в котором производная заменена конечно – разностным приближением. Этот метод является «двухшаговым» – для вычисления нового приближения к корню $x^{(k+1)}$ нужно знать два предыдущих приближения $x^{(k)}$, $x^{(k-1)}$. В данном методе для новое приближение к корню уранения $f(x)=0$ вычисляется по формуле:

$$x^{(k+1)}=x^{(k)}-\frac{x^{(k-1)}-x^{(k)}}{f\left(x^{(k-1)}\right)-f\left(x^{(k)}\right)}f\left(x^{(k)}\right).$$

Задание № 10

Тестовые значения: 1.85718, 4.5364

Идея метода обратной квадратичной интерполяции состоит в том, чтобы по ранее найденным трем приближениям $x^{(k-2)}$, $x^{(k-1)}$, $x^{(k)}$ к корню уравнения $y=f(x)$ найти следующее приближение $x^{(k+1)}$. Для этого в данном методе строится квадратичный многочлен, такой, что:

$$x^{(i)}=P_2\left(f\left(x^{(i)}\right)\right),$$

где $i=k-2, k-1, k.$ За очередное приближение к корню принимается $x^{(k+1)}=P_2(0)$. Метод называется трехшаговим, поскольку для получения очередного приближения нужно найти три предыдущих. Необхидимым условием построения многочлена $P_2(y)$ является различие значений $f\left(x^{(i)}\right)$. Коэфициенты a, b, c многочлена $P_2(y)=a\cdot y^2+b\cdot y+c$ получаются в результате решения системы линейных уравнений

$$\begin{cases} a\cdot\left(y^{(k-2)}\right)^2 & + & b\cdot y^{(k-2)} & + & c & = & x^{(k-2)} \\ a\cdot\left(y^{(k-1)}\right)^2 & + & b\cdot y^{(k-1)} & + & c & = & x^{(k-1)} \\ a\cdot\left(y^{(k)}\right)^2 & + & b\cdot y^{(k)} & + & c & = & x^{(k)} \end{cases}$$

в которой $y^{(i)}=f\left(x^{(i)}\right)$, $i=k-2, k-1, k$. После вычислений получим:

$$x^{(k+1)}=P_2(0)=x^{(k-2)}\frac{y^{(k-1)}\cdot y^{(k)}}{\left(y^{(k-2)}-y^{(k-1)}\right)\cdot\left(y^{(k-2)}-y^{(k)}\right)}+x^{(k-1)}\frac{y^{(k-2)}\cdot y^{(k)}}{\left(y^{(k-1)}-y^{(k-2)}\right)\cdot\left(y^{(k-1)}-y^{(k)}\right)}+$$
$$+x^{(k)}\frac{y^{(k-2)}\cdot y^{(k-1)}}{\left(y^{(k)}-y^{(k-2)}\right)\cdot\left(y^{(k)}-y^{(k-1)}\right)}$$

Метод обратной квадратичной интерполяции обладает локальной сходимостью и для начала его работы требуется задание трех «хороших» начальных приближений.

Задание № 11

Тестовое значение: $0.106675+2.6459\cdot i$

Для определения комплексных корней уравнения $f(z)=0$ можно воспользоваться методом Мюллера. Метод является трехшаговым и основан на примерно тех же идеях, что и метод обратной квадратичной интерполяции. В данном методе для поиска очередного приближения к корню необходимы три ранее найденные приближения $z^{(k-2)}$, $z^{(k-1)}$, $z^{(k)}$ и соответствующие значения функции $f\left(z^{(k-2)}\right)$, $f\left(z^{(k-1)}\right)$, $f\left(z^{(k)}\right)$. По этим трем точкам составляется единственная квадратичный полином $P_2(z)$, интерполирующий эти точки, т.е. такой, что $P_2\left(z^{(i)}\right)=f\left(z^{(i)}\right)$, для $i=k-2, k-1, k.$ За очередное приближение $z^{(k+1)}$ принимается тот из двух корней многочлена $P_2(z)$, который ближе расположен к $z^{(k)}$.
Коэффициенты c_1, c_2, c_3 многочлена $P_2(z)=c_1\cdot z^2+c_2\cdot z+c_3$ получаются в результате решения системы линейных уравнений

$$\begin{cases} c_1\cdot\left(z^{(k-2)}\right)^2 & + & c_2\cdot z^{(k-2)} & + & c_3 & = & y^{(k-2)} \\ c_1\cdot\left(z^{(k-1)}\right)^2 & + & c_2\cdot z^{(k-1)} & + & c_3 & = & y^{(k-1)} \\ c_1\cdot\left(z^{(k)}\right)^2 & + & c_2\cdot z^{(k)} & + & c_3 & = & y^{(k)} \end{cases}$$

в которой $y^{(i)}=f\left(z^{(i)}\right)$, $i=k-2, k-1, k$. После вычислений получим:

$$c_1 = \frac{y^{(k)}\cdot\left(z^{(k-2)}-z^{(k-1)}\right)+y^{(k-2)}\cdot\left(z^{(k-1)}-z^{(k)}\right)+y^{(k-1)}\cdot\left(z^{(k)}-z^{(k-2)}\right)}{\left(z^{(k-2)}-z^{(k-1)}\right)\cdot\left(z^{(k-2)}-z^{(k)}\right)\cdot\left(z^{(k-1)}-z^{(k)}\right)}$$

$$c_2=\frac{y^{(k)}\cdot\left(\left(z^{(k-1)}\right)^2-\left(z^{(k-2)}\right)^2\right)+y^{(k-1)}\cdot\left(\left(z^{(k-2)}\right)^2-\left(z^{(k)}\right)^2\right)+y^{(k-2)}\cdot\left(\left(z^{(k)}\right)^2-\left(z^{(k-1)}\right)^2\right)}{\left(z^{(k-2)}-z^{(k-1)}\right)\cdot\left(z^{(k-2)}-z^{(k)}\right)\cdot\left(z^{(k-1)}-z^{(k)}\right)}$$

$$c_3=\frac{y^{(k)}\cdot z^{(k-2)}\cdot\left(z^{(k-2)}-z^{(k-1)}\right)\cdot z^{(k-1)}+z^{(k)}\cdot\left(y^{(k-2)}\cdot z^{(k-1)}\cdot\left(z^{(k-1)}-z^{(k)}\right)+y^{(k-1)}\cdot z^{(k-2)}\cdot\left(z^{(k)}-z^{(k-2)}\right)\right)}{\left(z^{(k-2)}-z^{(k-1)}\right)\cdot\left(z^{(k-2)}-z^{(k)}\right)\cdot\left(z^{(k-1)}-z^{(k)}\right)}$$

Полученные формулы необходимо проверить.

Задание № 12

Корень уравнения локализован на отрезке $0\leqslant x<1$. Для $p_1=3\pi/4$ корень уравнения приближенно равен 0.838929.

Тема № 7. Многофайловые программы. Модули

На данном занятии необходимо познакомиться с некоторыми численными методами, применяемыми для решения обыкновенных дифференциальных уравнений первого порядка. Кроме того, нужно продолжить практику работы с проектами *Фортрана*, состоящими из нескольких файлов и изучить использование новой программной единицы *Фортрана* —модуля для хранения глобальных данных.

Список заданий

Задание № 1

Найти численное решение обыкновенного дифференциального уравнения

$$\frac{\mathrm{d}\,y(t)}{\mathrm{d}\,t}=y(t)-t^2+1$$

с заданной точностью ε на интервале $t\in[a,b]$ с заданным начальным условием $y(a)=\alpha$.

Для расчетов взять $\varepsilon=10^{-5}$, $a=0$, $b=2$, $\alpha=-0.5,-0.4,\ldots,0.5$. Во время каждого вычисления сравнить полученный численный разультат с «точным» (полученным аналитически) решением. С помощью программы *GnuPlot* построить общий график.

Необходимые численные методы, возможная структура расчетной программы и структура входных – выходных файлов приведены в *Рекомендациях по выполнению* данного задания.

Задание № 2

Для обыкновенного дифференциального уравнения

$$\frac{\mathrm{d}\,y(t)}{\mathrm{d}\,t}=y(t)-t^2+1$$

найти с заданной точностью ε такое начальное условие $y(0)$, чтобы на конце интервала интегрирования $(t=2)$ получить заданное значение функции $(y(2)=1)$. Получить решение обыкновенного дифференциального уравнения с заданным начальным условием, сравнить с аналитически полученным решением, построить график в программе *GnuPlot*.

Замечания по выполнению приведены в *Рекомендациях по выполнению* данного задания.

Задание № 3

С помощью разработанных на данном занятии процедур получить численное решение обыкновенного дифференциального уравнения

$$\frac{\mathrm{d}\,y(t)}{\mathrm{d}\,t}=\beta\cdot y(t)-t^2+1$$

с заданной точностью ε на интервале $t\in[a,b]$ с заданным начальным условием $y(a)=\alpha$. Для расчетов взять $\varepsilon=10^{-5}$, $a=0$, $b=2$, несколько значений α и β. Во время каждого вычисления сравнить полученный численный разультат с «точным» (полученным аналитически) решением. С помощью программы *GnuPlot* построить график полученного численного решения.

Замечания по выполнению приведены в *Рекомендациях по выполнению* данного задания.

Рекомендации по выполнению

Задание № 1

Основные замечания по решению обыкновенных дифференциальных уравнений

Для эффективного решения системы обыкновенных дифференциальных уравнений рекомендуется применять метод с переменным шагом интегрирования. Величина этого переменного шага будет выбираться самой программой (исходя из задаваемого пользователем начального значения) таким образом, чтобы удовлетворить некоторому критерию достижения требуемой точности приближенного решения.

Необходимо отметить, что формулы Рунге – Кутта очень хорошо приспособлены для интегрирования с переменным шагом, так как они позволяют легко менять шаг интегрирования и при этом не требуют никаких дополнительных вычислений и преобразований. Кроме того, если правая часть обыкновенного дифференциального уравнения не очень сложна, то методы типа Рунге – Кутта могут эффективно применяться для вычисления решения на всем интервале интегрирования.

Для решения обыкновенного дифференциального уравнения с контролем точности на каждом шаге можно воспользоваться методом Рунге – Кутта – Фельдберга. Данный метод позволяет эффективно оценивать локальную ошибку решения и широко используется для расчета электромагнитных полей в разных средах. Однако, необходимо отметить, что данный метод имеет ограниченную область устойчивости, поэтому он практически не пригоден для жестких уравнений.

Рассмотрим основные расчетные формулы. Для вычислений используется метод Рунге – Кутта пятого порядка:

$$\tilde{w}_{i+1}=w_i+\frac{16}{135}k_1+\frac{6656}{12825}k_3+\frac{28561}{56430}k_4-\frac{9}{50}k_5+\frac{2}{55}k_6,$$

для того, чтобы оценить локальную ошибку метода Рунге – Кутта четвертого порядка, заданного формулой:

$$w_{i+1}=w_i+\frac{25}{216}k_1+\frac{1408}{2565}k_3+\frac{2197}{4104}k_4-\frac{1}{5}k_5,$$

где

$$k_1=h\cdot f\left(t_i,w_i\right),$$

$$k_2=h\cdot f\left(t_i+\frac{h}{4},w_i+\frac{1}{4}k_1\right),$$

$$k_3=h\cdot f\left(t_i+\frac{3\cdot h}{8},w_i+\frac{3}{32}k_1+\frac{9}{32}k_2\right),$$

$$k_4=h\cdot f\left(t_i+\frac{12\cdot h}{13},w_i+\frac{1932}{2197}k_1-\frac{7200}{2197}k_2+\frac{7296}{2197}k_3\right),$$

$$k_5=h\cdot f\left(t_i+h,w_i+\frac{439}{216}k_1-8k_2+\frac{3680}{513}k_3-\frac{845}{4104}k_4\right),$$

$$k_6=h\cdot f\left(t_i+\frac{h}{2},w_i-\frac{8}{27}k_1+2k_2-\frac{3544}{2565}k_3+\frac{1859}{4104}k_4-\frac{11}{40}k_5\right).$$

Контрольный член, необходимый для оценки точности на шаге интегрирования может быть записан в виде:

$$E=\frac{1}{360}k_1-\frac{128}{4275}k_3-\frac{2197}{75240}k_4+\frac{1}{50}k_5+\frac{2}{55}k_6$$

и имеет пятый порядок точности.

Как большое преимущество данного метода нужно отметить то, что для каждого шага достаточно всего шести вычислений правой части уравнения f. Если работать стандартным образом: оценивать точность с помощью методов Рунге – Кутта пятого и шестого порядка, то для вычислений потребуется по крайней мере четыре вычисления f для метода четвертого порядка, и дополнительно шесть вычислений f для пятого порядка. Таким образом, всего потребуется не менее десяти вычислений правой части f. Кроме того, данный метод обычно точнее оценивает ошибку, чем метод Рунге - Кутта - Мерсона. На каждом шаге вычисляются указанные выше величины, оценивается ошибка и в зависимости от ее величины принимается решение о изменении шага интегрирования уравнения.
Рассмотрим один из возможных алгоритмов управления шагом интегрирования, основанный на данном методе решения дифференциального уравнения:

Входные данные

- ➔ границы отрезка интегрирования уравнения: a, b
- ➔ начальное значение: α
- ➔ точность вычисления: TOL
- ➔ максимально допустимая величина шага интегрирования: hMax
- ➔ минимально допустимая величина шага интегрирования: hMin

Выходные данные

- ➔ значение независимой переменной: t
- ➔ значение зависимой переменной: w (w аппроксимирует значение y(t))
- ➔ размер использованного шага интегрирования: h (или сообщение о том, что требуемый шаг меньше минимально допустимого)

```
Шаг 1. Присвоить:
     t ← a
     w ← α
     h ← hMax
     FLAG ← TRUE
     ВЫВОД(t, w)
Шаг 2. Пока (FLAG = TRUE) Выполнить Шаги 3 – 11
     Шаг 3. Присвоить:
```

$$K_1=h\cdot f(t,w);$$

$$K_2=h\cdot f\left(t+\frac{1}{4}\cdot h,w+\frac{1}{4}\cdot K_1\right);$$

$$K_3=h\cdot f\left(t+\frac{3}{8}\cdot h,w+\frac{3}{32}\cdot K_1+\frac{9}{32}\cdot K_2\right);$$

$$K_4=h\cdot f\left(t+\frac{12}{13}\cdot h,w+\frac{1932}{2197}\cdot K_1-\frac{7200}{2197}\cdot K_2+\frac{7296}{2197}\cdot K_3\right);$$

$$K_5=h\cdot f\left(t+h,w+\frac{439}{216}\cdot K_1-8\cdot K_2+\frac{3680}{513}\cdot K_3-\frac{845}{4104}\cdot K_4\right);$$

$$K_6 = h \cdot f\left(t + \frac{1}{2} \cdot h,\, w - \frac{8}{27} \cdot K_1 + 2 \cdot K_2 - \frac{3544}{2565} \cdot K_3 + \frac{1859}{4104} \cdot K_4 - \frac{11}{40} \cdot K_5\right).$$

Шаг 4. Присвоить: $\left(R = \frac{1}{4}\left|\widetilde{w}_{i+1} - w_{i+1}\right|\right)$

$$R = \frac{1}{h}\left|\frac{1}{360} \cdot K_1 - \frac{128}{4275} \cdot K_3 - \frac{2197}{75240} \cdot K_4 + \frac{1}{50} \cdot K_5 + \frac{2}{55} \cdot K_6\right|.$$

Шаг 5. Если $R \leqslant TOL$ То Выполнить Шаги 6 и 7:

Шаг 6. Приближенное решение принято. Присвоить:

$t = t + h;$

$$w = w + \frac{25}{216} \cdot K_1 + \frac{1408}{2565} \cdot K_3 + \frac{2197}{4104} \cdot K_4 - \frac{1}{5} \cdot K_5.$$

Шаг 7. ВЫВОД(t, w, h)

Шаг 8. Присвоить: $\delta = 0.84 \cdot (TOL/R)^{1/4}$.

Шаг 9. Вычисление новой длины шага h

Если $\delta \leqslant 0.1$ То Присвоить: h ← 0.1 h

Иначе Если $\delta \geqslant 4$ То Присвоить: h ← 4 h

Иначе Присвоить: h ← δ h.

Шаг 10. Если h > hMax То Присвоить: h ← hMax

Шаг 11. Если $t \geqslant b$ То Присвоить FLAG ← FALSE

Иначе Если $t+h>b$ То Присвоить: h ← b – t

Иначе Если h < nMin То

Начало БОК

Присвоить FLAG ← FALSE

ВЫВОД('Шаг должен быть меньше минимального')

(Процедура решения закончилась неудачей)

Шаг 12. СТОП

(Нормальное завершение процедуры решения дифференциального уравнения)

Аналитическое решение заданного уравнения

Аналитическое решение приведенного в задании обыкновенного дифференциального уравнения можно записать в виде:

$$y(t) = (\alpha - 1) \cdot e^t + (1+t)^2$$

здесь $\alpha = y(0)$. Изучить графики аналитического решения данного уравнения для разных начальных условий можно с помощью программы *GnuPlot*. Для этого в среде *GnuPlot* нужно определить функцию, соответствующую решению и затем построить ее график (см. **Рис. 7.1**).

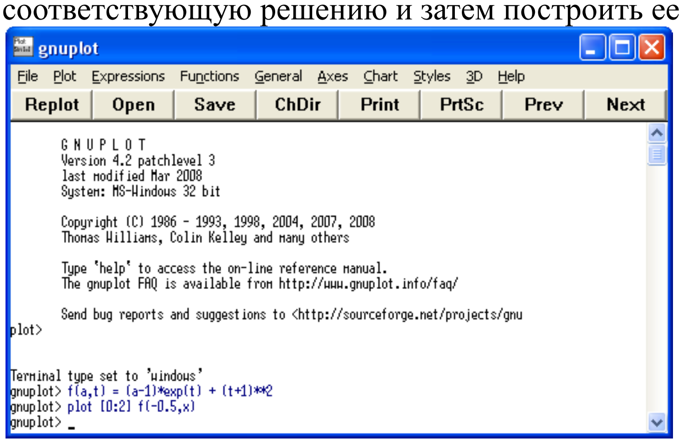

Рисунок 7.1. Определение функции и построение ее графика в программе *GnuPlot*.

Возможная структура входных / выходных данных программы

Входные данные:

Данные, считываемые программой из текстового файла (**`data.ini`**): начальное условие α, необходимое для численного интегрирования дифференциального уравнения.
Данные, получаемые программой с клавиатуры: границы отрзка интегрирования a, b, шаг *step*, с которым требуется получить численное решение.
Данные, указываемые в тексте программы: точность расчетов *tol*, максимально и минимально допустимые значение шага *hMax*, *hMin*.

Выходные данные:

Вывод на экран: полученное из файла значение начального условия α, численное решение обыкновенного дифференциального уравнения (значения t_i и $y(t_i)$ – значение независимой переменной и значение искомой функции в этой точке).
Вывод в текстовый файл (**`result.dat`**): значения t_i, $y(t_i)$ и величина «ошибки» – абсолютное значение разности численного и аналитического решений в заданной точке.

Возможная структура программы

Текст программы можно разместить в трех файлах с исходным кодом:

- файл **`ODE.f90`** — содержит функцию, определяющую обыкновенное дифференциальное уравнение:
  `function ode(t, y).`
  Здесь `t` — значение независимой переменной, `y` — значение функции в этой точке, результирующая переменная `ode` — значение производной в данной точке (правая часть приведенного обыкновенного дифференциального уравнения); Входные параметры: `t`, `y`.
- файл **`odeSolve.f90`** — содержит подпрограмму, которая находит численное решение обыкновенного дифференциального уравнения методом Рунге – Кутта – Фельдберга:
  `subroutine odeSolve(fun, xStart, xEnd, yStart, yEnd, tol, hMax, hMin).`
  Здесь `fun` — функция, определяющая обыкновенное дифференциальное уравнение, `xStart`, `xEnd` — начальное и конечное значения интервала, на котором ищется численное решение, соответственно (точки вывода), `yStart` — начальное значение функции в начале интервала поиска решения, `yEnd` — найденное значение функции в конечной точке интервала, `tol` — заданная точность расчетов, `hMax`, `hMin` — максимальное и минимальное значения шага интегрирования дифференциального уравнения на заданном участке, соответственно. Входные параметры: `fun`, `xStart`, `xEnd`, `yStart`, `tol`, `hMax`, `hMin`. Выходные параметры: `yEnd`;
- файл **`Task1.f90`** — основная программа, организующая всю работу; процедуры получения значений, необходимых для расчетов, из файла данных и из клавиатуры; функция, вычисляющая аналитическое решения заданного обыкновенного дифференциального уравнения.

Возможные скрипты программы GnuPlot

Для оперативного контроля за «качеством» решения обыкновенного дифференциального уравнения после каждого расчета следует запустить программу *GnuPlot* для вывода на экран ПК вычисленного решения и распределения ошибки – отличия численного решения от аналитического на всем протяжении интиервала интегрирования. Для этого можно взять за основу приведенный ниже скрипт:

```
reset
```

```
set nogrid  # без сетки на графике
set nokey   # без автоматической легенды
set tics out          #  штрихи у осей наружу
# заглавие графика
set title "ODE Solution" font "Arial Italic,16"
# построение графика в окне
plot 'result.dat' using 1:2 with lines linetype 1 linewidth 3
pause -1 "Press OK button to continue or Exit button to stop..."
# заглавие нового графика
set title "ODE Error" font "Arial Italic,16"
plot 'result.dat' using 1:3 with lines linetype 1 linewidth 3
pause -1 "Press OK button to continue or Exit button to stop..."
```

Все строки данного скрипта снабжены краткими пояснениями. Развернутую справку по использованным в скрипте командам можно получить в справочной системе программы *GnuPlot*.

Для построения финального графика по рассчитанным значениям за основу можно взять приведенный ниже скрипт:

```
reset       # сброс всех настроек в значение по умолчанию
set nogrid  # без сетки на графике
set nokey   # без автоматической легенды
set xrange [-0.1:2.1]  # диапазон x, отображаемый на графике
set yrange [-2.5:5.5]  # диапазон y, отображаемый на графике
set tics out          #  штрихи у осей наружу
set xtics border nomirror norotate 0.0, 0.5, 2.0
# оформление оси Ox
set ytics border nomirror norotate -2.0, 1.0, 5.0
# оформление оси Oy
set mxtics 10 # количество малых штрихов оси Ox
set mytics 2   # количество малых штрихов оси Oy
set label 1 "y(t)" at 0.0, 4.75 left norotate font "Arial, 22"
# текстовая метка
set label 2 "t" at 1.85, -2.0 left norotate font "Arial, 22"
# текстовая метка
# заглавие графика
set title "ODE Solution" font "Arial,24"
# построение графиков в окне
plot 'result_0_5.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result_0_4.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result_0_3.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result_0_2.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result_0_1.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result0_0.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result0_1.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result0_2.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result0_3.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result0_4.dat' using 1:2 with lines linetype 1 linewidth 3, \
     'result0_5.dat' using 1:2 with lines linetype 1 linewidth 3
pause -1 "Press OK button to continue or Exit button to stop..."
# настройка формата вывода
set terminal png font arial 20 size 800,600 \
```

```
xffffff x000000 x000000 x000000
# указание результирующего файла
set output 'fig1.png'
# в файле заглавие графика не нужно
set notitle
# далее - текстовые метки
set label 3 "1" at 2.02, -2.3 left norotate font "Arial, 20"
set label 4 "2" at 2.02, -1.5 left norotate font "Arial, 20"
set label 5 "3" at 2.02, -0.7 left norotate font "Arial, 20"
set label 6 "4" at 2.02,  0.1 left norotate font "Arial, 20"
set label 7 "5" at 2.02,  0.8 left norotate font "Arial, 20"
set label 8 "6" at 2.02,  1.5 left norotate font "Arial, 20"
set label 9 "7" at 2.02,  2.2 left norotate font "Arial, 20"
set label 10 "8" at 2.02, 3.0 left norotate font "Arial, 20"
set label 11 "9" at 2.02, 3.7 left norotate font "Arial, 20"
set label 12 "10" at 2.0, 4.5 left norotate font "Arial, 20"
set label 13 "11" at 2.0, 5.1 left norotate font "Arial, 20"
# перерисовка графика
replot
# Восстановление первоначальных настроек
set terminal windows
set output
pause -1 "Press any key to continue..."
reset
```

Все строки данного скрипта снабжены краткими пояснениями. Развернутую справку по использованным в скрипте командам можно получить в справочной системе программы *GnuPlot*.

Полученный график приведен на **Рис. 7.2.**

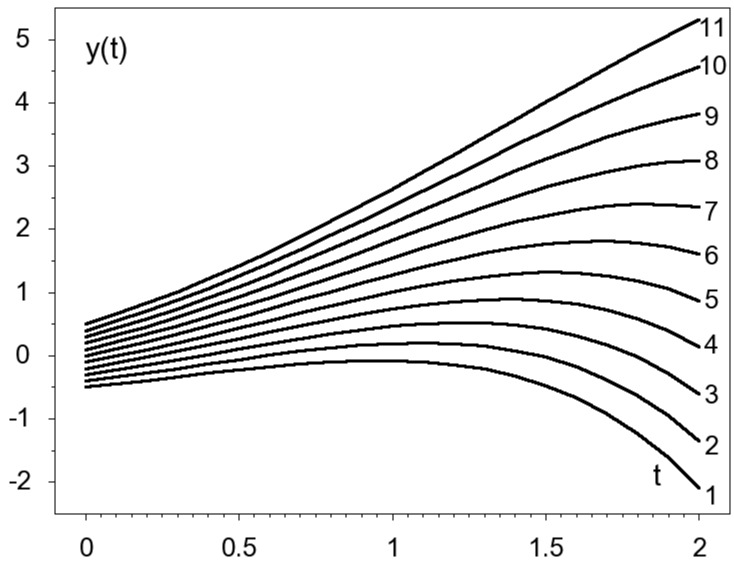

Рисунок 7.2. Результирующий график, построенный в программе *GnuPlot* по приведенному выше скрипту

Задание № 2

Основные замечания по решению задания

Для решения данного задания можно воспользоваться методом, похожим на метод «стрельбы», применяемый для решения граничных задач. Основная идея такого метода, в применении к решению данной задачи, заключается в угадывании значения начального условия α на левой границе a интервала интегрирования обыкновенного диференциального уравнения и затем, используя это угаданное значение, в численном решении заданного обыкновенного дифференциального уравнения от левой до правой границы интервала интегрирования. Если значение численно вычисленного решения $\tilde{y}(b,\alpha)$ на правой границе b интервала интегрирования не удовлетворяет заданному условию на этой границе β, то необходимо изменить наше угаданное значение для начального условия на левой границе и проинтегрировать дифференциальное уравнение (выстрелить) еще раз.

Значение вычисленного решения на правой границе интервала интегрирования является функцией начального условия на левой границе интервала интегрирования. Численное решение обыкновенного дифференциального уравнения параметрически зависит от этого начального условия. Следовательно, решая трансцендентное уравнение $\tilde{y}(b,\alpha)=\beta$ относительно начального условия α можно найти такое его значение, которое удовлетворит условию на правой границе интервала интегрирования. Далее, можно численно проинтегрировать обыкновенное дифференциальное уравнение, используя найденное начальное условие и, таким образом, поставленная задача будет решена.

Для решения трансцендентного уравнения можно воспользоваться разработанными на прошлом занятии процедурами решения нелинейных уравнений, например процедурой, реализующей метод бисекции (деления отрезка пополам).

Рекомендуется самостоятельно спроектировать структуру данной программы, взяв за основу программу из **Задания № 1**.

Возможные скрипты программы GnuPlot

Для построения результирующего графика по рассчитанному начальному значению за основу можно взять приведенный ниже скрипт:

```
reset          # сброс всех настроек в значение по умолчанию
set nogrid  # без сетки на графике
set nokey   # без автоматической легенды
set xrange [-0.1:2.1]  # диапазон x, отображаемый на графике
set yrange [-0.2:1.5]  # диапазон y, отображаемый на графике
set tics out          #  штрихи у осей наружу
set xtics border nomirror norotate 0.0, 0.5, 2.0
# оформление оси Ox
set ytics border nomirror norotate -0.2, 0.2, 1.4
# оформление оси Oy
set mxtics 10 # количество малых штрихов оси Ox
set mytics 2   # количество малых штрихов оси Oy
set label 1 "y(t)" at 0.0, 1.35 left norotate font "Arial, 22"
# текстовая метка
set label 2 "t" at 2.0, -0.125 left norotate font "Arial, 22"
# текстовая метка
# заглавие графика
set title "ODE Solution" font "Arial,24"
# построение графика в окне
plot 'result.dat' using 1:2 with lines linetype 1 linewidth 3
pause -1 "Press OK button to continue or Exit button to stop..."
# настройка формата вывода
```

```
set terminal png font arial 20 size 800,600 \
xffffff x000000 x000000 x000000
# указание результирующего файла
set output 'fig2.png'
# в файле заглавие графика не нужно
set notitle
# перерисовка графика
replot
# Восстановление первоначальных настроек
set terminal windows
set output
pause -1 "Press any key to continue..."
reset
```

Полученный график приведен на **Рис. 7.3.**

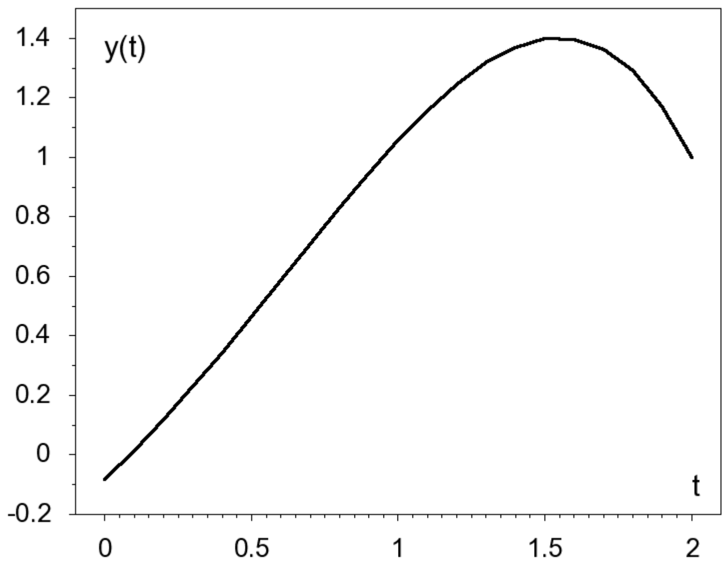

Рисунок 7.3. Результирующий график, построенный в программе *GnuPlot* по приведенному выше скрипту

Задание № 3

Основные замечания по решению задания

Для передачи полученного от пользователя параметра β в функцию, определяющую обыкновенное дифференциальное уравнение, а также для хранения интерфейсов к процедурам, можно воспользоваться новой программной единицей *Современного Фортрана* — модулем.

Рекомендуется самостоятельно спроектировать структуру данной программы, взяв за основу программу из **Задания № 1**.

Аналитическое решения данного обыкновенного дифференциального уравнения можно записать в виде:

$$y(t)=\frac{2+2\cdot t\cdot\beta+\left(t^2-1\right)\cdot\beta^2+e^{t\cdot\beta}\cdot\left(\beta^2\cdot(1+\alpha\cdot\beta)-2\right)}{\beta^3}$$