

**Лабораторные работы**  
**по дисциплине «Метрология и качество программного обеспечения»**

**Лабораторная работа № 1. Исследование качества программ сжатия данных.**

**Задание 1. Исследование свойств форматов сжатия графических данных.**

1. Подготовьте для экспериментов папку C:\Temp\Pictures.
2. Откройте графический редактор Paint (Пуск►Программы►Стандартные►Paint) и загрузите в него заранее подготовленный многоцветный рисунок .bmp.
3. Определите размер рисунка в пикселах (Рисунок►Атрибуты). Оцените теоретический размер рисунка в 24-разрядной палитре (3 байта на точку) по формуле

$S = M * N * 3$ , где  $S$  - размер файла с рисунком (байт);

$M$  - ширина рисунка (точек);

$N$  – высота рисунка (точек).

Запишите результат:  $S = \underline{\hspace{2cm}}$

4. Сохраните рисунок в отдельную папку, выбрав имя файла Test и назначив тип файла: 24-разрядный рисунок (.BMP).
5. Повторно сохраните рисунок, выбрав то же имя Test, но назначив тип файла .GIF. При сохранении произойдет потеря определенной части графической информации.
6. Восстановите рисунок, загрузив его из ранее сохраненного файла Test.bmp, и вновь сохраните его под тем же именем, но выбрав в качестве типа файла формат .JPEG.
7. Запустите программу Проводник. Откройте вашу папку с рисунками в режиме Таблица. Определите размеры файлов Test.bmp, Test.gif и Test.jpg. Определите коэффициент сжатия файлов ( $R$ ), взяв отношения размеров файлов к теоретической величине, полученной расчетным путем в п. 3. Результаты занесите в таблицу.

Формат файла	Размер файла (Кбайт)	Степень сжатия (%)
24-разрядный .BMP		
.GIF		
.JPG		

8. Ответьте на вопросы.
  - Что можно сказать о степени сжатия данных в формате .BMP?
  - Что можно сказать о степени сжатия данных в формате .JPEG?
9. В графическом редакторе Paint дайте команду создания нового документа Файл►Создать. Убедитесь в том, что полотно имеет размер 640x480. Если это не так, измените его размер командой Рисунок ► Атрибуты ► Ширина (Высота).
10. В качестве инструмента выберите Кисть. Задайте максимальный размер кисти.

Поочередно используя 8-10 разных красок, грубо закрасьте полотно.

11. Сохраните рисунок под именем Test\_1 в формате 24-разрядный рисунок (bmp).

12. Сохраните рисунок под именем Test\_1 в формате .GIF.

13. Восстановите рисунок из файла Test\_1 .bmp.

14. Сохраните рисунок под именем Test\_1 в формате .JPEG.

15. С помощью программы Проводник определите размеры сохраненных файлов и заполните таблицу.

Формат файла	Размер файла (Кбайт)	Степень сжатия (%)
24-разрядный .BMP		
.GIF		
.JPG		

16. Уточните ранее сделанный вывод о степени сжатия данных в формате JPEG, учитывая тот факт, что в первом случае изображение было многоцветным, а во втором случае оно имело небольшое количество цветовых оттенков.

17. Ответьте на вопросы.

• Какой формат графических данных из рассмотренных здесь наилучшим образом подходит для передачи цветного фотографического материала по каналам электронных сетей?

• Какой формат графических данных целесообразно использовать для передачи черно-белого фотографического материала по каналам электронных сетей?

• Какой формат наиболее благоприятен для передачи рисунков, имеющих малое количество цветовых оттенков (до 256)?

• Какой формат графических данных не годится для передачи информации по каналам электронных сетей, но удобен для хранения изображений, предназначенных для дальнейшей обработки?

## Задание 2. Исследование алгоритмов сжатия программы.

1. Подготовьте для экспериментов две папки Z:\Temp\input и Z:\Temp\Output.

2. Наполните экспериментальную папку Z:\Temp\input произвольным материалом в объеме 20-30 Мбайт.

3. Запустите диспетчер архивов.

4. Дайте команду File ► New Archive (Файл ► Создать архив) и создайте архив test\_1 в папке Z:\Temp\Output.

5. В диалоговом окне Add (Добавление в архив) выделите все папки, включаемые в архив (CTRL+A), убедитесь в том, что установлены флажки Include subfolders (Включая вложенные папки) и Save extra folder info (Сохранить структуру папок). В раскрывающемся списке Compression (Степень сжатия) выберите пункт None (Без сжатия).

6. Зафиксируйте время начала архивации по секундомеру и щелкните на команд

ной кнопке Add (Добавить).

7. Зафиксируйте время конца архивации и определите продолжительность процесса.
8. Аналогичным образом создайте файл Test\_2, выбрав режим сжатия Super fast (Сверхбыстрое сжатие) и замерив продолжительность процесса.
9. Создайте файл Test\_3 в режиме сжатия Fast (Быстрое сжатие) и замерьте продолжительность.
10. Создайте файл Test\_4 в режиме сжатия Normal (Обычное сжатие) и замерьте продолжительность.
11. Создайте файл Test\_5 в режиме сжатия Maximum (Максимальное сжатие), замерьте продолжительность процесса и результаты сведите в таблицу.

Режим сжатия	Исходный размер, Мбайт	Результирующий размер, Мбайт	Время, с	Степень сжатия, %	Эффективность метода
None					
Super fast					
Fast					
Normal					
Maximum					

Определите степень сжатия по формуле:

$$R = \frac{S_r}{S_i} 100\%, \text{ где } S_r - \text{размер результирующего файла, Мбайт; } S_i - \text{размер}$$

исходного файла, Мбайт;  $R$  - степень сжатия.

Эффективность метода оцените по абсолютной величине приращения степени сжатия к приращению времени сжатия:

$$E_i = \frac{|R_i - R_{i-1}|}{|t_i - t_{i-1}|}$$

12. Сделайте вывод о наиболее эффективном методе сжатия по критерию соотношения степени сжатия и расхода времени на операцию.

### **Задание 3. Исследование средств и методов повышения степени уплотнения дисков.**

Степень сжатия данных, реализуемая программой Сжатие данных (DriveSpace 3), невысока. Прежде всего, это связано с необходимостью сделать процесс записи/чтения данных быстрым и прозрачным для пользователя. Вместе с тем, в операционной системе Windows есть стандартные средства, позволяющие сделать «переупаковку» данных и значительно повысить коэффициент сжатия - речь идет о программе Агент сжатия. Правда, ее алгоритм работает столь медленно, что обслуживание уплотненных дисков нередко производят в автоматическом режиме в нерабочее время. Работа в автоматическом режиме обеспечивается благодаря наличию программ Назначенные задания и Мастер обслуживания.

1. Подготовьте экспериментальную папку, например Z:\Temp\Input.
2. Наполните ее произвольным набором графических файлов в формате .BMP.

Если под рукой нет такого набора, создайте произвольный рисунок в графическом редакторе Paint, сохраните его в виде файла в формате 24-разрядный рисунок BMP и размножьте путем копирования.

3. Вставьте уплотненный гибкий диск в дисковод A:

4. Откройте программу Проводник и скопируйте на уплотненный диск заранее подготовленный комплект файлов.

5. Запустите программу Сжатие диска (DriveSpace 3) и с помощью команды Диск ▶

Свойства определите достигнутую степень сжатия.

6. Дайте команду Дополнительно ▶ Степень сжатия и измените расчетный коэффициент сжатия в соответствии с замеренным текущим значением.

7. Дайте команду Диск ▶ Свойства и определите расчетную емкость уплотненного гибкого диска.

8. Результаты замеров занесите в таблицу.

9. Откройте программу Агент сжатия (Пуск ▶ Программы ▶ Стандартные ▶ Служебные ▶ Агент сжатия).

10. Щелчком на кнопке Настройка откройте диалоговое окно Настройка агента сжатия и включите в нем в нем режим сжатия HiPack (Улучшенное сжатие). Для этого включите переключатель Вообще не применять UltraPack (Высокое быстродействие) и в группе Применять HiPack для остальных файлов включите переключатель Да.

11. Закройте диалоговое окно настройки щелчком на кнопке ОК и запустите Агент сжатия щелчком на кнопке Пуск.

12. После того как Агент сжатия закончит переупаковку гибкого диска, откройте окно программы Сжатие диска (DriveSpace 3), определите достигнутую степень

сжатия (см. п. 5), измените расчетный коэффициент сжатия (см. п. 6), определите новую расчетную емкость гибкого диска (см. п. 8) и результаты замеров зане

сите в таблицу.

13. Откройте окно программы Агент сжатия, откройте диалоговое окно ее настройки (см. п. 10) и включите режим сжатия UltraPack (Наилучшее сжатие).

14. Закройте окно настройки щелчком на кнопке ОК, запустите Агент сжатия щелчком на кнопке Пуск.

15. После того как Агент сжатия закончит работу, выполните замеры необходимых параметров (см. п.п. 5,6,8) и результаты занесите в таблицу.

Метод сжатия	Исходный размер массива	Коэффициент уплотнения	Расчетная емкость гибкого диска
Обычный			
HiPack (Улучшенный)			
UltraPack (Наилучший)			

16. Проанализируйте результаты исследовательской работы, сделайте выводы и дайте рекомендации.

17. По окончании работы попытайтесь отформатировать уплотненный гибкий диск стандартными средствами: откройте окно Мой компьютер, щелкните правой кнопкой мыши на значке дисководов A: и выберите в контекстном меню пункт Форматировать. Убедитесь в том, что форматирование невозможно из-за того, что диск уплотнен.

18. По указанию от операционной системы используйте для форматирования уплотненного диска программу Сжатие диска (DriveSpace 3). Самостоятельно найдите средства для выполнения этой операции.

## **Лабораторная работа № 2. Исследование качества программ, написанных на языках программирования высокого уровня.**

Решить приведённые по вариантам задачи на 3 трех языках программирования и оценить качество сделанных программ, а также провести сравнительную характеристику.

Вариант распределяется по списку студентов группы и выбираются из приведённого ниже списка:

1. Составить алгоритм и программу, которые из вводимого целого числа в переменную  $X$  печатают отдельно каждую цифру числа, не используя встроенных функций.
2. Составить алгоритм и программу, которые для последовательности  $5^n/n!$ ,  $n = 0, 1, 2, \dots$  вычисляет максимальный элемент, его номер и выводят их на экран.
3. Составить алгоритм и программу, которые в одну и ту же переменную вводят  $N$  действительных чисел. Определить сумму чисел, порядковые номера которых являются четными числами и среднее арифметическое чисел, порядковые номера, которых являются нечетными числами.
4. Составить алгоритм и программу, которые вводят границы натуральных двузначных чисел и из этих чисел печатают только те, цифры которых являются соседними в натуральном ряду. Произвести подсчет напечатанных чисел и определить, сколько в них четных и нечетных.
5. Составить алгоритм и программу, которые в числовую переменную вводят отличные от нуля целые числа. Количество чисел заранее не известно. Определить и вывести на экран, какие два числа (положительные или отрицательные) были введены последними, а также определить их сумму.
6. Составить алгоритм и программу (без использования массивов), которые в числовую переменную  $N$  вводят ненулевые целые числа. Количество вводимых чисел заранее неизвестно. Определить и вывести сообщение о том, сколько раз в этой последовательности сменился знак, а также - определить максимальное число отрицательных чисел, идущих в этой последовательности подряд.
7. Составить алгоритм и программу, которые в числовые переменные вводят

два натуральных числа, являющиеся числителем и знаменателем простой дроби, и получают новые значения этих чисел, сократив дробь до несократимой дроби. Предусмотреть проверку ввода исходных данных.

8. Составить алгоритм и программу, которые выводят в строку все натуральные простые числа, не превосходящие заданное натуральное трехзначное число  $M$ , а также находят количество этих чисел и их сумму.
9. Составить алгоритм и программу, которые в числовую переменную  $N$  вводят натуральное число, не превышающее 3000, и определить, является ли оно «совершенным». Выдать на экран соответствующее сообщение. «Совершенным» называется число, равное, сумме всех своих делителей, исключая само число. Например:  $27 = 1+2+4+7+14$ .
10. Составить алгоритм и программу, которые находят минимальное число слагаемых в сумме членов ряда, при котором эта сумма станет больше 120, и вывести найденную сумму, последнее слагаемое и его номер.

$$11+122+1233+1234+\dots$$

11. Составить алгоритм и программу вычисления суммы таких членов последовательности  $a_i = x/(1*2*3* \dots *i)$ ,  $i = 1, 2, \dots$ , значения которых больше 0.01. Вводимое значение  $0 < x < 1$ .
12. Составить алгоритм и программу вычисления произведения  $k$  натуральных чисел, кратных  $m$  и не превышающих  $n$ . Известно, что  $m, n$  - натуральные.
13. Составить алгоритм и программу, которые в числовые переменные вводят два натуральных числа  $X$  и  $N$  и находят ближайшее к  $X$  и расположенное в пределах от  $X$  до  $N$  натуральное число, кратное  $N$ , а также - произведение его цифр. Результаты вывести на экран. В программе предусмотреть проверку правильности исходных данных.

$$\left(1 + \frac{1}{1}\right) * \left(1 + \frac{1}{2}\right) * \left(1 + \frac{1}{3}\right) * \dots$$

14. Составить алгоритм и программу расчета величины  $Y$

$$Y = \frac{2 * x^2}{1} - \frac{4 * x^4}{9} + \frac{6 * x^6}{11} - \frac{8 * x^8}{13} + \dots + \frac{14 * x^{14}}{19}$$

15. Составить алгоритм и программу расчета 10-ти членов ряда:
16. Составить алгоритм и программу нахождения минимального натуральною числа  $N$ , при котором выполняется условие:

$$1^3 + 2^3 + 3^3 + \dots + n^3 > 300$$

17. Составить алгоритм и программу, которые вводят целое значение  $X$  ( $10 < X < 20$ ) и находят сумму цифр целой части числа, являющеюся произведением 12-ти членов последовательности:

$$X \left(X - \frac{1}{2}\right) \left(X - \frac{1}{2} - \frac{1}{2}\right) \left(X - \frac{1}{2} - \frac{1}{2} - \frac{1}{4}\right) \dots$$

18. Составить алгоритм и программу, которые в одну и ту же переменную вводят вещественные значения  $X$ , удовлетворяющие условию  $1 < |X| < 3$  (количество вводимых  $X$  заранее неизвестно) и для каждого  $X$  находят значения 15-ти членов ряда и их сумму

$$1 + \frac{x}{1 \cdot 2} + \frac{x^2}{1 \cdot 2 \cdot 5} - \frac{x^3}{1 \cdot 2 \cdot 5 \cdot 6} - \frac{x^4}{1 \cdot 2 \cdot 5 \cdot 6 \cdot 9} - \frac{x^5}{1 \cdot 2 \cdot 5 \cdot 6 \cdot 9 \cdot 10} + \dots$$

19. Составить алгоритм и программу, которые в одну и ту же переменную вводят вещественные значения  $X > 0$  (количество вводимых  $X$  заранее неизвестно) и для каждого  $X$  находят значение произведения членов последовательности, умножая их до тех пор, пока очередное слагаемое не станет меньше 0,001

$$x, \frac{x^r}{r!}, \frac{x^5}{5!}, \frac{x^y}{y!}, \frac{x^9}{9!}, \dots$$

20. Составить алгоритм и программу, которые вычисляют сумму членов последовательности, до тех пор, пока модуль разности между её предыдущим и текущим членами остается больше 0,001. Кроме суммы вывести на экран значение последнего члена ряда и его номер.

$$1, \frac{1}{4}, \frac{1}{12}, \frac{1}{32}, \frac{1}{80}, \dots$$

21. Составить алгоритм и программу, которые находят максимальное количество членов ряда, при котором сумма ее членов остается меньше 20, и выводят на экран значение последнего слагаемого и его сумму:

$$7 + \frac{y}{2*1} + \frac{y}{3*2*1} + \frac{y}{4*3*2*1} + \dots$$

22. Составить алгоритм и программу, которые находят сумму членов ряда:

$$\frac{x^r}{1^r} + \frac{x^f}{2^r} + \frac{x^f}{3^r} + \dots + \frac{x^{10}}{9^r}$$

23. Составить алгоритм и программу, которые вычисляют произведение членов ряда:

$$P = x * \frac{x^5}{33} * \frac{x^y}{60} * \frac{x^9}{96} * \dots * \frac{x^{10}}{256}$$

24. Составить алгоритм и программу, которые вычисляют сумму членов ряда:

$$P = x + \frac{3 * x^f}{8} - \frac{6 * x^y}{10} + \frac{9 * x^{10}}{12} - \frac{12 * x^{12}}{14} + \dots + \frac{21 * x^{12}}{20}$$

### Выполнение работы.

Программы составить на Pascal, Visual Basic и Delphi. Оценить качество программ и занести результаты оценки в следующую таблицу:

Таблица 1.

Язык программирования	Исходный размер программы, Мбайт	Время выполнения, сек	Размер программы в текстовом эквиваленте,	Эффективность метода
Pascal				50 %
Visual Basic				
Delphi				

Эффективность метода оцените по абсолютной величине приращения степени сжатия к приращению времени сжатия:

$$E_i = \frac{|R_i - R_{i-1}|}{|t_i - t_{i-1}|},$$

$R_i$  - размер результирующего файла, Мбайт;  $R_{i-1}$  - размер исходного файла, Мбайт;  $t_i$  - время выполнения результирующего файла,  $t_{i-1}$  - время выполнения исходного файла.

В качестве исходного файла выбрать файл программы написанной на языке программирования Pascal.

Расчёты привести в развернутом виде.

Лабораторную работу представить в виде распечаток текстов самих программ, блок-схем алгоритмов, результатов выполнения программ и выполненных расчетов.

### **Лабораторная работа № 3. Исследование зависимости точности и времени вычисления функций от методов их вычисления на персональном компьютере.**

**Цель работы:** Оценка погрешности и времени вычисления функций при использовании разных методов их нахождения.

В некоторых задачах скорость выполнения часто встречающихся операций является более важной характеристикой программы, чем точность результата. При этом целесообразно достичь некоторого компромисса между двумя указанными характеристиками. К упомянутым задачам можно отнести алгоритмы оптимизации, машинной графики и другие. В них приходится вычислять алгебраические, тригонометрические и другие функции.

Можно считать, что нахождение значений функций с помощью стандартных средств (модуля System или библиотек программ), обеспечивает максимальную точность. Будем считать такой подход эталонным. Наиболее распространенными приближенными методами являются:

- вычисление функций с помощью степенных рядов;
- использование таблиц значений функций.

Точность методов зависит в первом случае от количества членов ряда ( $n1$ ), а во втором – от размера таблицы (количества значений  $n2$  в ней).

В работе необходимо оценить погрешность вычисления заданной преподавателем функции по формуле

$$\delta = \text{Max} (F_{\text{эталон}}(x) - F_{\text{приближен}}(x)),$$

где  $F_{\text{эталон}}(x)$  – значение функции, найденное эталонным методом,  $F_{\text{приближен}}(x)$  - ее значение, полученное приближенным методом.

Кроме того, нужно определить время нахождения функции тремя описанными методами и построить графики зависимостей погрешности и времени от величин  $n1$  и  $n2$ . При этом должно быть исследовано поведение

функций для разных значений аргумента  $x$ , задаваемых преподавателем (например, от  $x_{\text{нач}}$  до  $x_{\text{кон}}$  с шагом  $hx$ ), и построены графики функции  $Y = F(x)$  на заданном интервале. Интервал значений аргумента для каждого варианта приведен в таблице 1 Приложения 2.

В лабораторной работе необходимо составить процедуры приближенного вычисления заданной преподавателем функции обоими методами по следующим алгоритмам.

- 1. Вычисление с помощью рядов.** Рассмотрим на примере функции  $\sin x$  из таблицы 1 Приложения 2. Формула для вычисления функции приведена там же. Она имеет вид:

$$y = F(x) = x - x^3 / 3! + x^5 / 5! - x^7 / 7! + \dots$$

Метод 1 можно реализовать с помощью следующего алгоритма

1. Построить эталонный график функции  $F_{\text{эталон}}(x)$ , меняя  $x$  от  $x_{\text{нач}}$  до  $x_{\text{кон}}$  с шагом  $hx = (x_{\text{кон}} - x_{\text{нач}}) / 50$ .
2. Для  $n1$  от 3 до 10 выполнить
  - 2.1. Положить  $x = x_{\text{нач}}$
  - 2.2. Пока  $x \leq x_{\text{кон}}$  выполнить
    - 2.2.1. Предыдущее =  $x$ .
    - 2.2.2. Следующее =  $-x^3 / 6$ .
    - 2.2.3.  $Y =$  Предыдущее.
    - 2.2.4.  $n = 1$ .
    - 2.2.5. Для  $k$  от 1 до  $n1 - 1$  выполнить
      - 1) Следующее =  $-$  Предыдущее \*  $x^2 / ((n + 1) * (n + 2))$
      - 2)  $Y = Y +$  Следующее
      - 3)  $n = n + 2$ .
    - 2.2.6.  $F1_{n1}(x) = Y$ .
    - 2.2.7.  $x = x + hx$ .
    - 2.2.8. Вывести точку на графике  $F1$  с координатами, определяемыми  $x$  и  $F1_{n1}(x)$ , и соединить ее с предыдущей точкой.
    - 2.2.9. Определить значение погрешности для очередного  $n1$  по формуле:  $\delta_{n1} = F_{\text{эталон}}(x) - F1_{n1}(x)$ .

**Примечание.** Для построения графика функции эталонным методом (см. п. 1 алгоритма) интервал значений  $x$  разбивается на 50 – 100 отрезков, на границах которых вычисляются значения  $y$  с помощью стандартных функций языка программирования. Эти величины определяют координаты  $x$  и  $y$

эталонного графика. Точки графика соединяются прямыми с помощью графических процедур используемого языка программирования.

**2. Нахождение значений функции с помощью таблиц.** Выполняется следующим образом.

Интервал значений  $x$  последовательно делится на 2, 3, 4 и т.д. отрезка (по количеству величин  $n2$  в таблице). Значения функции на границах отрезков вычисляются с помощью эталонного метода (стандартных функций языка программирования). При этом все промежуточные значения считаются лежащими на отрезке прямой, соединяющей указанные границы (как показано на рис. 2.1).

Погрешность вычисления функции можно оценить абсолютной величиной разности между координатой  $y_{отр}(x)$  промежуточной точки  $x$  отрезка и эталонным значением функции в этой точке, т. е. по следующей формуле:

$$\delta = |y_{эт}(x) - y_{отр}(x)| = |y_{эт}(x) - (y2 - y1) * h / (x2 - x1) - y1|,$$

где  $y_{отр}(x) = (y2 - y1) * h / (x2 - x1) + y1$  - координата промежуточной точки  $x$  отрезка, отстоящей от точки  $x1$  на величину  $h$  (рис.).

С учетом полученных соотношений метод 2 можно реализовать с помощью следующего алгоритма.

1. Построить эталонный график функции  $F_{эталон}(x)$ , меняя  $x$  от  $x_{нач}$  до  $x_{кон}$  с шагом  $hx = (x_{кон} - x_{нач}) / 50$ .

2. Для  $n2$  от 3 до 10 выполнить

2.1. Положить  $X = x_{нач}$

2.2. Шаг  $hx = (x_{кон} - x_{нач}) / n2$

2.3. Вычислить  $y1_{эт}(X)$

2.4. Положить  $y1 = y1_{эт}(X)$

2.5. Положить  $x1 = X$ .

2.6. Пока  $X \leq x_{кон}$  выполнить

2.6.1. Вычислить  $X = X + hx$ .

2.6.2. Вычислить  $y2_{эт}(X)$

2.6.3. Положить  $y2 = y2_{эт}(X)$

2.6.4. Положить  $x2 = X$ .

2.6.5. Провести отрезок от точки  $(x1, y1)$  до точки  $(x2, y2)$ .

2.6.6. Меняя  $x$  от  $x1$  до  $x2$  с шагом  $hx / 10$ , вычислять погрешности  $\delta$  по формуле (2.1), и найти максимальное значение  $\delta_{max}$

2.6.7. Положить  $x1 = X$ .

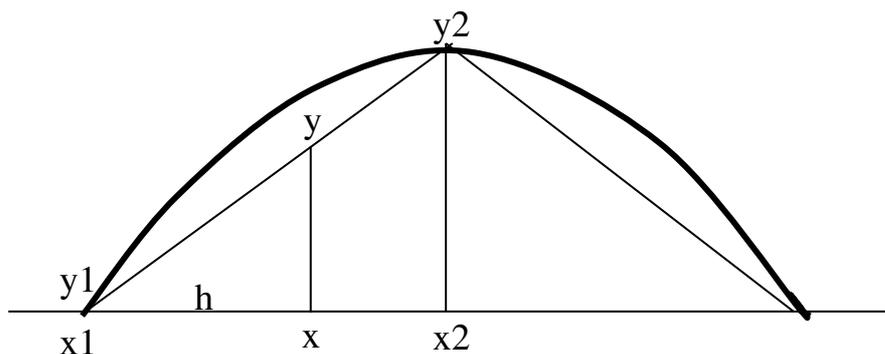


Рис. 2.1. Аппроксимация функции  $F(x)$  при табличном методе

**Примечание.** Построение графика функции эталонным методом (см. п. 1 алгоритма) выполняется так же, как для метода 1, разбивая интервал значений  $x$  на 50 – 100 отрезков.

### ***Порядок выполнения работы***

1. Пользуясь описанными выше алгоритмами, разработать и отладить программу вычисления заданной преподавателем функции тремя методами: эталонным и двумя приближенными.
2. Построить кривые  $F_{\text{эталон}}(x)$  и  $F_{\text{приближен}}(x)$  на заданном интервале значений  $x$  (см. Приложение 2).
3. Построить кривые времени вычисления функции разными методами в зависимости от числа членов ряда  $n1$  или значений в таблице  $n2$ . Время вычисления определять с помощью системного таймера. Для уменьшения погрешности оценки времени процедуры нахождения функции выполнять 10 000 - 100 000 раз.
4. Построить кривые  $\delta_{\text{max}}$  для вычисления функции разными методами в зависимости от числа членов ряда  $n1$  или значений в таблице  $n2$ .

### ***Содержание отчета о выполненной работе***

Отчет о работе должен содержать:

- 1) Вид заданной преподавателем функции  $F(x)$  и формулы, используемые для ее приближенного вычисления;
- 2) Максимальные значения погрешностей для каждого метода;
- 3) Графики  $F_{\text{эталон}}(x)$  и  $F_{\text{приближен}}(x)$  на заданном интервале значений  $x$ , которые демонстрируются на экране монитора;
- 4) Графики времени вычисления функции разными методами, а также  $\delta_{\text{max}}$  в зависимости от числа членов ряда  $n1$  или значений в таблице ( $n2$ ), которые демонстрируются на экране монитора.

## Литература

1. Куликовский К.Л., Купер В.Я. Методы и средства измерений. - М.: Высшая школа, 1987.
2. Современные высокопроизводительные компьютеры. Информационно-аналитический обзор. М.: Центр Информационных технологий, - 1995.

## ПРИЛОЖЕНИЯ Исследуемые функции

№	Эталонная функция	Интервал значений $x$	Формула для вычисления с помощью ряда
1	$\text{Sin}(x)$	$0 \leq x \leq \pi$	$F(x) = x - x^3 / 3! + x^5 / 5! - x^7 / 7! + \dots$
2	$\text{Cos}(x)$	$0 \leq x \leq \pi / 2$	$F(x) = 1 - x^2 / 2! + x^4 / 4! - x^6 / 6! + \dots$
3	$e^x$	$-1 \leq x \leq 1$	$F(x) = 1 + x + x^2 / 2! + x^3 / 3! + x^4 / 4! + \dots$
4	$e^{-x}$	$-1 \leq x \leq 1$	$F(x) = 1 - x + x^2 / 2! - x^3 / 3! + x^4 / 4! - \dots$
5	$\text{Ln}(1+x)$	$-1 < x < 1$	$F(x) = x - x^2 / 2 + x^3 / 3 - x^4 / 4 + \dots$
6	$\text{Ln}(1-x)$	$-1 < x < 1$	$F(x) = -x - x^2 / 2 - x^3 / 3 - x^4 / 4 - \dots$
7	$\text{Ln}(x)$	$0 < x < 2$	$F(x) = (x-1) + (x-1)^2 / 2 + (x-1)^3 / 3 + \dots$
8	$a^x$	$-1 \leq x \leq 1, a=2$	$F(x) = 1 + x \cdot \ln(a) + (x \cdot \ln(a))^2 / 2! + (x \cdot \ln(a))^3 / 3! + \dots$
9	$\text{ArcTg}(x)$	$1 \leq x \leq 2$	$F(x) = \pi / 2 - 1/x + 1/3 \cdot x^3 - 1/5 \cdot x^5 + \dots$
10	$\text{ArcCTg}(x)$	$1 \leq x \leq 2$	$F(x) = 1/x - 1/3 \cdot x^3 + 1/5 \cdot x^5 - \dots$