Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра прочности летательных аппаратов

“УТВЕРЖДАЮ”
ДЕКАН ФЛА
д.т.н. Саленко С. Д.
“___ ”______________ ______ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Практикум по алгоритмизации и программированию

Образовательная программа: 15.03.03 Прикладная механика, профиль: Динамика и прочность

Курс: 2, семестры : 3 4

Факультет летательных аппаратов,

		Семестр	
№	**Вид деятельности**	**3**	**4**
1	Всего зачетных единиц (кредитов)	3	2
2	Всего часов	108	72
3	Всего занятий в контактной форме, час.	61	42
4	Лекции, час.	0	0
5	Практические занятия, час.	0	0
6	Лабораторные занятия, час.	54	36
7	из них в активной и интерактивной форме, час.	0	0
8	Аттестация, час.	2	2
9	Консультации, час.	5	4
10	Самостоятельная работа, час.	47	30
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ	РГЗ
12	Вид аттестации	ДЗ	З

Новосибирск 2017

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 15.03.03 Прикладная механика

ФГОС введен в действие приказом №220 от 12.03.2015 г. , дата утверждения: 16.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 15.03.03 Прикладная механика

Рабочая программа обсуждена на заседании кафедры

ПЛА, протокол заседания кафедры №5/1 от 20.06.2017

Утверждена на совете факультета летательных аппаратов, протокол № 5 от 21.06.2017

Программу разработал:

доцент, к.т.н. Белоусов А. И.

Заведующий кафедрой:

профессор, д.т.н. Пустовой Н. В.

Ответственный за образовательную программу:

профессор Левин В. Е.

1. Внешние требования

Таблица 1.1

***Компетенция ФГОС*: ОПК.10 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; *в части следующих результатов обучения*:**
у2. уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ
у4. владеть персональным компьютером как средством управления информацией
у5. уметь использовать специализированные программные средства при решении профессиональных задач
у6. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов
***Компетенция ФГОС*: ПК.4 готовность выполнять научно-исследовательские работы в области прикладной механики с использованием современных вычислительных методов, высокопроизводительных вычислительных систем и наукоемких компьютерных технологий, широко распространенных в промышленности систем мирового уровня, и экспериментального оборудования для проведения механических испытаний; *в части следующих результатов обучения*:**
з2. знать структуру и область применения современных языков программирования
у3. уметь программировать на одном из языков высокого уровня
***Компетенция ФГОС*: ПК.5 способность составлять описания выполненных научно-исследовательских работ и разрабатываемых проектов, обрабатывать и анализировать полученные результаты, готовить данные для составления отчетов и презентаций, написания докладов, статей и другой научно-технической документации; *в части следующих результатов обучения*:**
у1. уметь работать со специальными программами для составления отчетов и презентаций
***Компетенция ФГОС*: ПК.6 способность применять программные средства компьютерной графики и визуализации результатов научно-исследовательской деятельности, оформлять отчеты и презентации, готовить рефераты, доклады и статьи с помощью современных офисных информационных технологий, текстовых и графических редакторов, средств печати; *в части следующих результатов обучения*:**
з1. знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей
з2. знать основные программные средства компьютерной графики и визуализации результатов научно-исследовательской деятельности
з3. знать основные специализированные программные средства для решения профессиональных задач

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
Практикум по алгоритмизации и программированию	
ПК.4.з2 знать структуру и область применения современных языков программирования	
1.знать структуру и область применения современных языков программирования	Лабораторные работы; Самостоятельная работа
ПК.4.у3 уметь программировать на одном из языков высокого уровня	
2.уметь программировать на одном из языков высокого уровня	Лабораторные работы; Самостоятельная работа
3.уметь использовать языки и системы программирования для решения профессиональных задач	Лабораторные работы; Самостоятельная работа
ПК.5.у1 уметь работать со специальными программами для составления отчетов и презентаций	
4.уметь работать со специальными программами для составления отчетов и презентаций	Лабораторные работы; Самостоятельная работа

ПК.6.з1 знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей	
5.знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей	Лабораторные работы; Самостоятельная работа
ПК.6.з2 знать основные программные средства компьютерной графики и визуализации результатов научно-исследовательской деятельности	
6.знать основные программные средства компьютерной графики и визуализации результатов научно-исследовательской деятельности	Лабораторные работы; Самостоятельная работа
ПК.6.з3 знать основные специализированные программные средства для решения профессиональных задач	
7.знать основные специализированные программные средства для решения профессиональных задач	Лабораторные работы; Самостоятельная работа
ОПК.10.у2 уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	
8.уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	Лабораторные работы
ОПК.10.у4 владеть персональным компьютером как средством управления информацией	
9.владеть персональным компьютером как средством управления информацией	Лабораторные работы
ОПК.10.у5 уметь использовать специализированные программные средства при решении профессиональных задач	
10.уметь использовать специализированные программные средства при решении профессиональных задач	Лабораторные работы
ОПК.10.у6 уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	
11.уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	Лабораторные работы

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 3				
Дидактическая единица: ДЕ 1				
1. Аппроксимация функций	0	10	6, 7	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.
2. Расчет стержня сложного поперечного сечения на растяжение	0	14	1, 11, 2, 3, 4, 5, 6, 7	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.

3. Методы решения задач на собственные значения	0	10	10, 2, 4, 7, 8	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.
4. Математическая модель и программа решения задачи по теории колебаний.	0	10	1, 2, 3, 4, 6, 9	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.
5. Комплексные величины при выполнении расчетов	0	10	1, 11, 2, 3	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.
Семестр: 4				
Дидактическая единица: ДЕ 2				
6. Расчет шарнирно-опертой балки	0	12	11, 3, 7	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.
7. Расчет статически неопределимой системы	0	12	11, 3, 7	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.
8. Расчет ломаного бруса	0	12	11, 3, 7	Составление алгоритма решения поставленной задачи. Разработка и ввод программ (Excel, MathCAD, FORTRAN). Отладка программ. Сопоставление результатов. Оформление отчета.

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 3				
1	РГЗ	1, 2, 3	45	5
, подробная информация приведена в приложениях №1 №2 №3 №4 №5 №6 №8 : Информатика и программирование : методические указания для проведения лабораторных работ по курсу "Информатика" для 1 курса ФЛА / Новосиб. гос. техн. ун-т ; [сост.: М. А. Леган и др.]. - Новосибирск, 2003. - 59 с. : ил.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2003/2558.rar				

2	Подготовка к аттестации	1, 2, 3	2	0
: Информатика и программирование : методические указания для проведения лабораторных работ по курсу "Информатика" для 1 курса ФЛА / Новосиб. гос. техн. ун-т ; [сост.: М. А. Леган и др.]. - Новосибирск, 2003. - 59 с. : ил.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2003/2558.rar				
Семестр: 4				
1	РГЗ	1, 2, 3, 5, 6, 7	28	4
, подробная информация приведена в приложении №9 : Информатика и программирование : методические указания для проведения лабораторных работ по курсу "Информатика" для 1 курса ФЛА / Новосиб. гос. техн. ун-т ; [сост.: М. А. Леган и др.]. - Новосибирск, 2003. - 59 с. : ил.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2003/2558.rar				
2	Подготовка к аттестации	1, 2, 3, 4, 5, 6, 7	2	0
: Информатика и программирование : методические указания для проведения лабораторных работ по курсу "Информатика" для 1 курса ФЛА / Новосиб. гос. техн. ун-т ; [сост.: М. А. Леган и др.]. - Новосибирск, 2003. - 59 с. : ил.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2003/2558.rar				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	e-mail; Портал НГТУ
Консультирование	e-mail; Портал НГТУ
Контроль	e-mail; Портал НГТУ
Размещение учебных материалов	e-mail; Портал НГТУ; ЭБС

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставлять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Максимальный балл
Семестр: 3	
РГЗ:	80
Зачет:	20
Контролирующие материалы - список вопросов	
Семестр: 4	
РГЗ:	80
Зачет:	20
Контролирующие материалы - список вопросов	

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	**Результаты обучения**	**Формы контроля**	
		Защита РГЗ	**Зачет**
ОПК.10	у2. уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	+	+
	у4. владеть персональным компьютером как средством управления информацией		+
	у5. уметь использовать специализированные программные средства при решении профессиональных задач	+	+
	у6. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	+	+
ПК.4	з2. знать структуру и область применения современных языков программирования		+
	у3. уметь программировать на одном из языков высокого уровня	+	+
ПК.5	у1. уметь работать со специальными программами для составления отчетов и презентаций	+	+
ПК.6	з1. знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей		+
	з2. знать основные программные средства компьютерной графики и визуализации результатов научно-исследовательской деятельности		+
	з3. знать основные специализированные программные средства для решения профессиональных задач		+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Плис А. И. Mathcad. Математический практикум для инженеров и экономистов : учебное пособие для вузов / А. И. Плис, Н. А. Сливина. - М., 2003. - 655 с. : ил.

2. Информатика. Базовый курс : [учебное пособие для втузов] / под ред. С. В. Симоновича. - СПб. [и др.], 2008. - 639 с. : ил.. - На тит. л.: Издательская программа 300 лучших учебников для высшей школы.

3. Корнеев И. К. Информационные технологии : учебник / И. К. Корнеев, Г. Н. Ксандопуло, В. А. Машурцев ; Гос. ун-т упр. - Москва, 2007. - 222 с. : ил.

4. Информатика [Электронный ресурс] : учебник / Новосиб. гос. техн. ун-т ; [О. К. Альсова и др.]. - Новосибирск, 2012. - 1 электрон. опт. диск (CD-ROM). - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000175426. - Загл. с этикетки диска.

5. Краинский И. Word 2007 : популярный самоучитель / И. Краинский. - СПб. [и др.], 2008. - 240 с. : ил.

6. Информатика и программирование : методические указания для проведения лабораторных работ по курсу "Информатика" для 1 курса ФЛА / Новосиб. гос. техн. ун-т ; [сост.: М. А. Леган и др.]. - Новосибирск, 2003. - 59 с. : ил.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2003/2558.rar

Дополнительная литература

1. Плис А. И. Лабораторный практикум по высшей математике : учебное пособие для втузов / А. И. Плис, Н. А. Сливина. - М., 1983. - 208 с.

2. Бартеньев О. В. Современный Фортран / О. В. Бартеньев. - М., 2000. - 448 с.

3. Бартеньев О. В. Современный Фортран : [учебное пособие] / О. В. Бартеньев. - М., 1998. - 397 с. : табл.

Интернет-ресурсы

1. ЭБС НГТУ : http://elibrary.nstu.ru/

2. ЭБС «Издательство Лань» : https://e.lanbook.com/

3. ЭБС IPRbooks : http://www.iprbookshop.ru/

4. ЭБС "Znanium.com" : http://znanium.com/

5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Виноградов А. В. Информатика [Электронный ресурс] : учебно-методическое пособие / А. В. Виноградов, Ю. О. Поляков ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000162183. - Загл. с экрана.

8.2 Специализированное программное обеспечение

1 Microsoft Office

2 Microsoft Windows

9. Материально-техническое обеспечение

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Компьютеры объединены в локальную сеть с выходом в Internet

РАБОТА N1. АППРОКСИМАЦИЯ ТАБЛИЧНО ЗАДАННЫХ ФУНКЦИЙ МЕТОДОМ НАИМЕНЬШИХ КВАДРАТОВ.

1. Метод наименьших квадратов.

Обычно набор экспериментальных данных получается с некоторой погрешностью. В этом случае необходимо провести аппроксимирующую кривую, которая не проходит через экспериментальные точки, но в то же время отражает исследуемую зависимость, сглаживает возможные выбросы, возникшие за счет погрешности эксперимента.

Обозначим узлы исходной таблицы данных через x_i, где $0\leq i \leq n$ номер узла. Считаем известными значения экспериментальных данных в узловых точках $f(x_i)=f_i$. Введем непрерывную функцию $\varphi(x)$ для аппроксимаци дискретной зависимости $f(x_i)$. В узлах таблицы функции $\varphi(x)$ и $f(x)$ будут отличаться на величину $\varepsilon_i=\varphi(x_i)-f(x_i)$. Отклонения ε_i могут принимать положительные и отрицательные значения. Чтобы не учитывать знаки, возведем каждое отклонение в квадрат и просуммируем квадраты отклонений по всем узлам

$$Q=\sum_{i=0}^{n}\varepsilon_i^2=\sum_{i=0}^{n}[\varphi(x_i)-f(x_i)]^2 . \qquad (1.1)$$

Метод построения аппроксимирующей функции $\varphi(x)$,следуеющий из условия минимума величины Q , называется методом наименьших квадратов (МНК).

В качестве аппроксимирующей функции в зависимости от характера точечного графика выбирается прямая, экспонента и т.д. Когда вид приближающей функции установлен, задача сводится только к отысканию значений параметров.

Рассмотрим метод нахождения параметров приближающей функции в общем виде на примере функции с тремя параметрами:

$$y=F(x,a,b,c). \qquad (1.2)$$

Итак, имеем: $F(x_i,a,b,c)=\bar{y}_i, i=1,2,...,n.$ Сумма квадратов разностей соответствующих значений f и F будет иметь вид:

$$\sum_{i=1}^{n}[y_i-F(x_i,a,b,c)]^2=\Phi(a,b,c). \quad (1.3)$$

Эта сумма является функцией Φ(a, b, c) трех переменных . Задача сводится к отысканию ее минимума. Используем необходимое условие экстремума:

$$\frac{d\Phi}{da}=0,\frac{d\Phi}{db}=0,\frac{d\Phi}{dc}=0, \quad (1.4), \qquad \text{т.е.}$$

$$\sum_{i=1}^{n}[y_i-F(x_i,a,b,c)]F'_a(x_i,a,b,c)=0,$$

$$\sum_{i=1}^{n}[y_i - F(x_i, a, b, c)]F_b'(x_i, a, b, c) = 0,$$

$$\sum_{i=1}^{n}[y_i - F(x_i, a, b, c)]F_c'(x_i, a, b, c) = 0, \qquad (1.5)$$

Решив систему (1.5) относительно параметров *a, b,* и *c*, мы и получим конкретный вид искомой функции *F*(*x, a, b, c*). Как видно из рассмотренного примера, изменение количества параметров выразиться лишь в изменении количества уравнений в системе (1.5).

Найденные значения функции *F*(*x, a, b, c*) в точках $x_1, x_2, \ldots, x_n$ будут отличаться от табличных значений $y_1, y_2, \ldots, y_n$. Значениия разностей

$$y_1 - F(x_1, a, b, c) = \varepsilon_1 (i = 1,2,\ldots,n) \qquad (1.6)$$

называются *отклонениями* (или *уклонениями*) измеренных значений *y* от вычесленных по формуле (1.6) в соответствии с исходными данными. Следовательно можно найти сумму квадратов отклонений,

$$\sigma = \sum_{i=1}^{n} \varepsilon^2, \qquad (1.7)$$

которая в соответствии с принципом наименьших квадратов для заданного вида приближающей функции (и найденных значений параметров *a, b,* и *c*) должна быть наименьшей.

Задание к работе: массив данных (задается преподавателем) аппроксимировать уравнением прямой линией :

$$F(x, a, b) = ax + b.$$

После нахождения неизвестных коэффициентов аппроксимирующей функции, отбраковать из массива точки отклонение которых больше полутора (двух) среднеквадратичных и аппроксимировать оставшийся массив. Постротить график аппроксимирующей функции и исходые точки с помощью графических пакетеов или FORTRAN- графики.

РАБОТА N2. ИНТЕРПОЛЯЦИЯ ФУНКЦИЙ КУБИЧЕСКИМ СПЛАЙНОМ

В данном разделе рассматриваются вопросы интерполяции функции одной переменной, заданной табличным способом , т.е. когда по известным значениям некоторой функции $f_i=f(x_i)$, заданным в несовпадающих точках $x_i(i= 0,1,\ldots,N)$, называемых узлами, требуется определить приближенное значение *f(x)* в произвольной точке *x*. В дальнейшем будем предполагать, что значения x_i являются возрастающими, непрерывный аргумент $x \in [x_0, x_N]$, т.е. функция *f=f(x)* непрерывна вместе со своими производными до порядка n включительно.

Наибольшее распространение среди интерполирующих сплайнов получил сплайн третьей степени (кубический сплайн), среди аппроксимирующих - локальные сплайны (или B-сплайны) [2]. Кубический сплайн проходит через все узловые точки, при этом все точки “влияют” на любой участок интерполирующей функции. Вид локального сплайна на каком-либо участке определяют только соседние узловые тоочки, и он в общем случае не проходит через узлы.

Рассмотрим способы интерполяции таблично заданных функций с помощью кубического сплайна и локальных B-сплайнов (Работа N3).

2.1. Кубический сплайн

На каждом из интервалов $[x_{n-1}, x_n], 1 \le n \le N$ таблично заданной функции $fi(i = 0,1,..., N)$ кубический сплайн представляется полиномом третьей степени

$$S_n(x) = \sum_{\ell=0}^{3} a_{en}(x_n - x)^e, n = 1,2,..., N \qquad (2.1.1)$$

Из требования интерполяции, а также равенства самой сплайн-функции $S_n(x)$ (отсутствие разрывов линии), ее первых (отсутствие изломов) и вторых (отсутствие скачков кривизны) производных на границах соседних интервалов можно получить *4N-2* уравнений для *4N* неизвестных:

$$\begin{aligned}
&a_{on} = f_n, n = 0,1,..., N \\
&a_{on} + a_{1n}h_n + a_{2n}h_n^2 + a_{3n}h_n^3 = f_{n-1}, n = 2,3,..., N \\
&a_{on} + 2a_{2n}h_n + 3a_{3n}h_n^2 = a_{1,n-1}, n = 2,3,..., N \\
&2a_{2n} + 6a_{3n}h_n = 2a_{2,n-1}, n = 2,3,..., N
\end{aligned} \qquad (2.1.2)$$

где $h_n = x_n - x_{n-1}$.

Для замыкания системы (2.1.2) задаются значения вторых производных (кривизны) на концах отрезка интерполяции :

$$S_1^2(x_o) = C_o, S_n^{(2)}(x_n) = C_1$$

или

$$\begin{aligned}
&3a_{31}h_1 + a_{21} = C_o / 2 \\
&a_{2N} = C_1 / 2
\end{aligned} \qquad (2.1.3)$$

Из (2.1.2) и (2.1.3) получаются выражения для определения неизвестных:

$$a_{3n} = \frac{a_{2,n-1} - a_{2n}}{3h_n}, n = 1,2,..., N \qquad (2.1.4)$$

где $a_{2o} = c_o / 2$,

$$a_{1n} = -\frac{h_n}{3}(a_{2,n-1} + 2a_{2n}) + \frac{f_{n-1} - f_n}{h_n} \qquad (2.1.5)$$

$$h_n a_{2,n-1} + 2(h_n + h_{n+1})a_{2n} + h_{n+1}a_{2,n+1} = F_n \qquad (2.1.6)$$

где $F_n = 3\left[\frac{f_{n-1} - f_n}{h_n} - \frac{f_n - f_{n+1}}{h_{n+1}}\right], a_{2o} = c_o / 2, a_{2N} = c_1 / 2$

Трехдиагональная система уравнений (2.1.6) решается методом прогонки [1] .

Рассмотрим суть этого метода на примере системы с трехдиагональной матрицей общего видда

$$A_j Z_{j-1}+C_j Z_j+B_j Z_{j+1}=F_j\ ,\ j=1,2,\dots,N-1\ .\quad (2.1.7)$$
$$Z_0=K_1\ ,\ \ Z_N=K_2$$

Подставим значение Z_0 в первое уравнение систнмы (2.1.7). Получим

$$C_1Z_1+B_1Z_2=F_1-A_1K_1$$

или

$$Z_1=\alpha_1 Z_2+\beta_1 \qquad \alpha_1=-\frac{B_1}{C_1}\ ,\ \beta_1=\frac{F_1-A_1K_1}{C_1} \quad .$$

Полученное Z_1 подставим в следующее уравнение и т. д. Пусть найдено

$$Z_{k-1}=\alpha_{k-1}Z_k+\beta_{k-1}\ ,\ \ k<N-1\ .$$

Подставив его в k-е уравнение получим

$$A_k(\alpha_{k-1}Z_k+\beta_{k-1})+C_kZ_k+B_kZ_{k+1}=F_k$$

или

$$Z_k=\alpha_k Z_{k+1}+\beta_k\ ;$$
$$\alpha_k=-\frac{B_k}{C_k+A_k\alpha_{k-1}};\beta_k=\frac{A_k\beta_{k-1}-F_k}{C_k+A_k\alpha_{k-1}} \quad .$$

При этом значения α_0 и β_0 определяются из левого граничного условия $Z_0=K_1$, для чего достаточно положить $\alpha_0=0$ *и* $\beta_0=K_1$. Таким образом все α_k и β_k определяются из рекурентных соотношений. Процесс их вычисления называется прямой прогонкой, а сами коэффициенты называются прогоночными. При этом левое граничное решение ‘‘перегоняется’’ на правую границу.

После вычисления прогоночных коэффициентов, используется правое граничное условие $Z_N=K_2$ и определюется все неизвестные значения (обратная прогонка)

$$Z_k=\alpha_k Z_{k+1}+\beta_k\ ,\ к=N-1,\ N-2,\ \dots\ ,\ 1.$$

После того как найдены все a_{2n} , из (2.1.2), (2.1.3), (2.1.4), (2.1.5) определяются все остальные неизвестные. Имея массив коэффициентов (по четыре коэффициента для каждого участка), можно определить значение функции по формуле (2.1.1) в любой точке.

Задание к лабораторной работе: для функции $C_x=f(M)$, заданной в виде графика (см. задание к лабораторной N6), выбрать необходимое колличество узловых точек. Исходные данные представить в виде таблицы. В процессе решения найти значения в промежуточных точках, а в тестовом примере и в узлах таблицы. (Обратите внимание, что программа должна содержать блок, определяющий номер участка по его координате *x*). По полученным данным построить график и проанализировать поведение интерполируемой функции. Массив коэффициентов сохранить для использования при выполнении работы N6.

РАБОТА N3. АППРОКСИМАЦИЯ ФУНКЦИЙ ЛОКАЛЬНЫМ В-СПЛАЙНОМ .

Вопросы аппроксимации таблично заданной функции с помощью локальных В-сплайнов рассмотрим для единичного отрезка $x \in [0,1]$ и постоянного шага таблицы *h=1/N.* Заметим, что произвольный отрезок $[x_o, x_k]$ легко приводится к единичному заменой переменной

$$\bar{x} = \xi = \frac{x - x_o}{x_k - x_o}.$$

Рассмотрим два В-сплайна третьей степени второго $B_2(x)$ и четвертого $B_4(x)$ порядков точности на гладких функциях, в основу которой положен так называемый стандартный В-сплайн третьей степени

$$B_S(\bar{x}) = \begin{cases} \frac{2}{3} - x^2 + \frac{1}{2}|x|^3, & |x| \le 1 \\ \frac{1}{6}(2 - |x|)^3, & 1 \le |x| \le 2, \\ 0 & |x| \rangle 2 \end{cases} \qquad (3.1)$$

Локальный сплайн второго порядка точности имеет вид :

$$B_2(\bar{x}) = \sum_{n=-1}^{N+1} f_n B_s\left(\frac{\bar{x} - nh}{h}\right) \qquad (3.2)$$

Смысл формулы (3.2) заключается в том, что значение функции при какой-либо координате x определяется как сумма значений *fn* в нескольких соседних узлах, взятых с весовым множителем *Bs* , определяемым формулой (3.1) (*Bs* меньше единицы, причем уменьшается по мере удаления от рассматриваемой точки).

Значения функции *f* в дополнительных узлах (так называемых фиктивных точках) таблицы получаются линейной экстраполяцией

$$f_{-1} = 2f_o - f_1, f_{N+1} = 2f_N - f_{N-1} \quad .$$

Локальный сплайн четвертого порядка точности имеет вид:

$$B_4(\bar{x}) = \sum_{n=-1}^{N=+1} \frac{8f_n - f_{n+1} - f_{n-1}}{6} B_s\left(\frac{\bar{x} - nh}{h}\right) \qquad (3.3)$$

где: $f_{-1} = 4f_o - 6f_1 + 4f_2 f_3,$

$f_{-2} = 10f_o - 20f_1 + 15f_2 - 4f_3,$

$f_{N-1} = 4f_n - 6f_{n-1} + 4f_{N-2} - f_{N-3},$

$f_{N-2} = 10f_n - 20f_{n-1} + 15f_{N-2} - 4f_{N-3}$,

- значения функции в дополнительных точках.

Задание к лабораторной работе: проинтерполировать локальными сплайнами второго и четвертого порядка точности функцию C_x=*f(M),* заданную в виде графика (см. задание к лабораторной N6). Выбрать необходимое колличество узловых точек, ввод данных осуществить в виде таблицы. В процессе решения найти значения в промежуточных точках, а в тестовом примере в узлах таблицы. По полученным данным построить график, на котором будут исходные узловые точки, а также функции полученные с помощью В-сплайнов второго и четвертого порядков точности. Проанализировать поведение функций.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра прочности летательных аппаратов

“УТВЕРЖДАЮ”
ДЕКАН ФЛА
д.т.н., профессор С.Д. Саленко
“___ ”_______________ _______ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Практикум по алгоритмизации и программированию
Образовательная программа: 15.03.03 Прикладная механика, профиль: Динамика и прочность

Новосибирск 2017

1. **Обобщенная структура фонда оценочных средств учебной дисциплины**

Обобщенная структура фонда оценочных средств по **дисциплине** Практикум по алгоритмизации и программированию приведена в Таблице.

Таблица

Формируемые компетенции	**Показатели сформированности компетенций (знания, умения, навыки)**	**Темы**	**Этапы оценки компетенций**	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	**Промежуточная аттестация (экзамен, зачет)**
ОПК.10 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	у2. уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	Методы решения задач на собственные значения	Лабораторные работы 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ОПК.10	у4. владеть персональным компьютером как средством управления информацией	Математическая модель и программа решения задачи по теории колебаний.	Лабораторные работы 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ОПК.10	у5. уметь использовать специализированные программные средства при решении профессиональных задач	Методы решения задач на собственные значения	Лабораторные работы 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ОПК.10	у6. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	Комплексные величины при выполнении расчетов Расчет ломаного бруса Расчет статически неопределимой системы Расчет стержня сложного поперечного сечения на растяжение Расчет шарнирно-опертой балки	Лабораторные работы 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ПК.4/НИ готовность выполнять научно-исследовательские работы в области прикладной	з2. знать структуру и область применения современных языков программирования	Комплексные величины при выполнении расчетов Математическая модель и программа решения задачи по теории колебаний. Расчет стержня сложного		Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра

механики с использованием современных вычислительных методов, высокопроизводительных вычислительных систем и наукоемких компьютерных технологий, широко распространенных в промышленности систем мирового уровня, и экспериментального оборудования для проведения механических испытаний		поперечного сечения на растяжение		
ПК.4/НИ	у3. уметь программировать на одном из языков высокого уровня	Комплексные величины при выполнении расчетов Математическая модель и программа решения задачи по теории колебаний. Методы решения задач на собственные значения Расчет ломаного бруса Расчет статически неопределимой системы Расчет стержня сложного поперечного сечения на растяжение Расчет шарнирно-опертой балки	РГЗ 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ПК.5/НИ способность составлять описания выполненных научно-исследовательских работ и разрабатываемых проектов, обрабатывать и анализировать полученные результаты, готовить данные для составления отчетов и презентаций, написания докладов, статей и другой научно-технической документации	у1. уметь работать со специальными программами для составления отчетов и презентаций	Математическая модель и программа решения задачи по теории колебаний. Методы решения задач на собственные значения Расчет стержня сложного поперечного сечения на растяжение	РГЗ 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ПК.6/НИ способность применять программные средства компьютерной графики и визуализации результатов научно-исследовательской	з1. знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей	Расчет стержня сложного поперечного сечения на растяжение	РГЗ 3,4 семестра	Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра

деятельности, оформлять отчеты и презентации, готовить рефераты, доклады и статьи с помощью современных офисных информационных технологий, текстовых и графических редакторов, средств печати				
ПК.6/НИ	з2. знать основные программные средства компьютерной графики и визуализации результатов научно-исследовательской деятельности	Математическая модель и программа решения задачи по теории колебаний. Расчет стержня сложного поперечного сечения на растяжение		Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра
ПК.6/НИ	з3. знать основные специализированные программные средства для решения профессиональных задач	Методы решения задач на собственные значения Расчет ломаного бруса Расчет статически неопределимой системы Расчет стержня сложного поперечного сечения на растяжение Расчет шарнирно-опертой балки		Зачет, вопросы 1-26 3-го семестра, 1-21 4-го семестра

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по **дисциплине** проводится в 3 семестре - в форме дифференцированного зачета, в 4 семестре - в форме зачета, который направлен на оценку сформированности компетенций ОПК.10, ПК.4/НИ, ПК.5/НИ, ПК.6/НИ.

Зачет проводится в устной форме, по билетам.

Форма проведения зачета описана отдельно.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 3 семестре обязательным этапом текущей аттестации являются расчетно-графические задания (работы) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

В 4 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.10, ПК.4/НИ, ПК.5/НИ, ПК.6/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы недостаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Задача №1

ЦЕНТРАЛЬНОЕ РАСТЯЖЕНИЕ И СЖАТИЕ ПРЯМОГО БРУСА.

ЗАДАНИЕ: вычислить таблицы значений и построить эпюры продольных сил N, нормальных напряжений σ и перемещений u поперечных сечений прямого бруса (рис. 1).

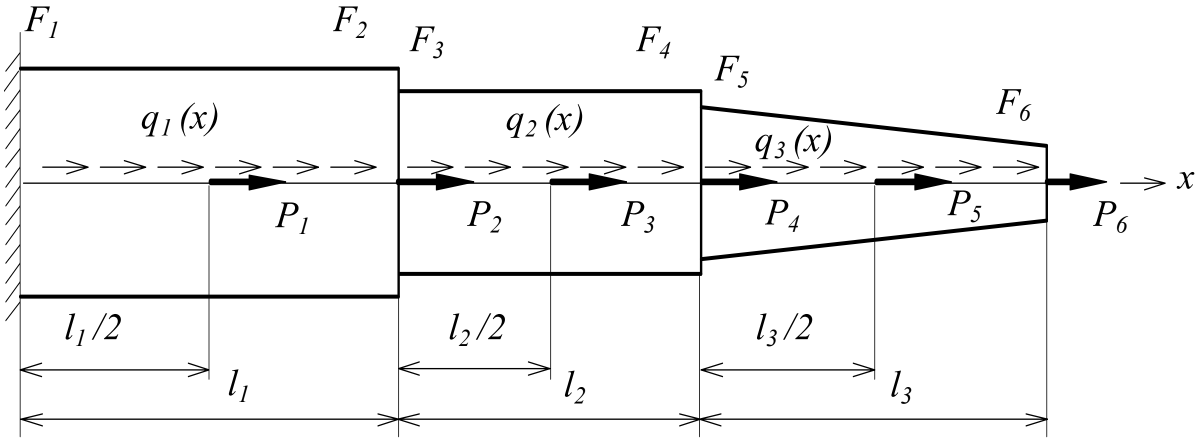


Рис. 1

Геометрически брус можно разделить на три участка. На i-ом участке (i =1, 2, 3) длиной l_i заданы F_{2i+1} , F_{2i} - площади левого и правого сечений участка бруса (на каждом участке принят линейный закон измененения площади по длине); P_{2i+1} , P_{2i} - сосредоточенные силы, приложенные в соответствии со схемой (рис. 1); $q_i\,(x)$ - продольная распределенная нагрузка, изменяющаяся по линейному закону (заданы k_i - тангенс наклона в уравнении прямой и b_i - значение q_i на левой границе участка).

Для вычисления значений N, σ, u необходимо воспользоваться следующими соотношениями:

$$N(x) = \int_x^{l_1+l_2+l_3} q(x)dx + \sum P_j \qquad (1)$$

В сумме (1) учитываются все сосредоточенные силы, приложенные на участке от x до l_1+l_2+l_3 .

$$\sigma(x) = N(x) / F(x) \qquad (2)$$

$$u\,(x) = \frac{1}{E}\int_0^x \sigma\,(x)d\,x \qquad (3)$$

Для решения задачи требуется разбить брус на m силовых участков (в общем случае $m = 6$) и на каждом участке силовом вычислить требуемые величины. Решение представить в виде таблицы и эпюр. На каждом силовом участке рассчитать не менее трех точек.

Для определения варианта задания порядковый номер по групповому журналу перевести в двоичный код и дополнить нулями до 6-ти позиций (например №3=000011). В зависимости от значений в каждой позиции (0 или 1) по таблице определить исходные данные. (E=1.1·10^8 кН / м^2)

№ позиции	1-ая			2-ая			3-ья		
значения в позиции	l_1 м	$F_1 \cdot 10^4$ м^2	$F_2 \cdot 10^4$ м^2	l_2 м	$F_3 \cdot 10^4$ м^2	$F_4 \cdot 10^4$ м^2	l_3 м	$F_5 \cdot 10^4$ м^2	$F_6 \cdot 10^4$ м^2
0	0.5	5	3	0.4	4	3	0.5	3	3
1	0.3	4	4	0.7	5	3	0.4	4	2

№ позиции	4-ая				5-ая				6-ая			
значения в позиции	P_1 кН	P_2 кН	k_1 кН/м^2	b_1 кН/м	P_3 кН	P_4 кН	k_2 кН/м^2	b_2 кН/м	P_5 кН	P_6 кН	k_3 кН/м^2	b_3 кН/м
0	0	10	5	2	0	-5	-4	-4	0	20	-5	-3
1	10	-10	0	0	-10	20	0	5	15	-5	0	0

При решении задачи возможны многие варианты программ. Рассмотрим на примере упрощенной схемы бруса (рис. 2) алгоритм одного из них.

ПОСТАНОВКА ЗАДАЧИ

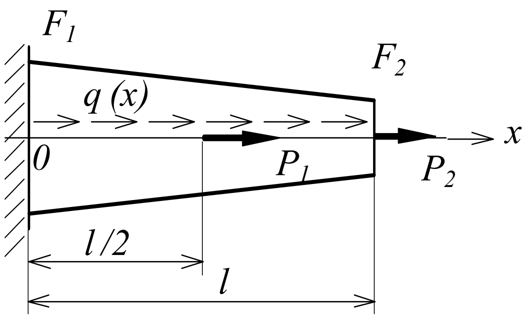


Рис. 2

Исходные данные: l = 1м , F_1= 4·10^{-4} м^2, F_2 = 2·10^{-4} м^2, P_1 = 10 Кн , P_2 = -20 Кн , k = 5 Кн / м^2, b = -5 Кн / м .

Уравнение изменения площади поперечного сечения бруса F(x) запишется в виде

$$F(x) = -2 \cdot 10^{-4} x + 4 \cdot 10^{-4}$$

Закон изменения распределенной нагрузки

$$q(x) = 5x - 5$$

Так как при $x = l / 2$ в точке приложения силы P_1 продольная сила N терпит разрыв, брус необходимо разбить на два силовых участка: 1-ый участок $0 < x < l / 2$; 2-ой участок $l / 2 < x < l$. При этом в точке $x = l / 2$ величины N и σ необходимо вычислить дважды: слева и справа от точки.

АЛГОРИТМ РЕШЕНИЯ.

Введем в рассмотрение массив $X(6)$, элементы которого числено равны координатам точек в которых вычисляются N, σ, u. При трех точках на каждом участке и равномерной сетке получим:

$$x_1 = 0, \quad x_2 = l / 4, \quad x_3 = l / 2, \quad x_4 = l / 2, \quad x_5 = 3l / 4, \quad x_6 = l .$$

Для вычислений и хранения усилий, напряжений и перемещений в точках x_i, будем использовать массивы $N(6)$, $\sigma(6)$, $u(6)$.

Вычисление усилий методом сечений удобно начинать со свободного конца бруса. Из расчетной схемы видно, что

$$N_6 = P_2 .$$

Для N_5 из (1) имеем:

$$N_5 = N_6 + \int_{x_5}^{x_6} q(x)dx$$

Вычисляя последний интеграл по формулам трапеций с использованием значений функции $q(x)$ только на концах отрезка интегрирования (т. к. $q(x)$ - линейная функция, то результат будет точным), получим

$$N_5 = N_6 + [q(x_6) + q(x_5)]\, h / 2$$

где $h = x_6 - x_5$ - шаг интегрирования. В рассматриваемом случае для равномерности сетки независимо от пределов интегрирования шаг постоянен и равен $h = l / 4$.

Для N_4 аналогично получим формулу

$$N_4 = N_5 + [q(x_5) + q(x_4)]\, l / 8$$

В точке разрыва имеем: $N_3 = N_4 + P_1$

и далее: $N_2 = N_3 + [q(x_3) + q(x_2)]\, l / 8$

$$N_1 = N_2 + [q(x_2) + q(x_1)]\, l / 8$$

Если увеличить длину массива N на единицу и ввести значение $N_7 = 0$, то вместо $N_6 = P_2$ получим $N_6 = N_7 + P_2$.

Тогда можно независимо от количества силовых участков и точек разбиения внутри участков записать следующие рекуррентные формулы для определения усилий в поперечных сечениях бруса:

$N_{m\cdot k+1} = 0$ - значение усилий за пределом бруса;

$N_{j\cdot k} = N_{j\cdot k+1} + P_j$, $j = m, m-1, \ldots, 1$ - при переходе через j-ый стык (стык j-го и j+1 -го силовых участков);

$N_i = N_{i+1} + [q(x_{i+1}) + q(x_i)]\, h / 2$, $i = j\cdot k + 1, \ldots, (j - 1)\cdot k + 1$ - для точек внутри j-го силового участка.

Здесь: m - число участков; k - количество точек на участке; h - шаг разбиения. Если при разбиении используется неравномерная сетка, то $h_i = x_{i+1} - x_i$.

Зная усилия и напряжения (последние определяются в соответствии с формулой (2)), можно вычислить перемещения в контрольных точках по формуле (3). При интегрировании по формуле трапеций (в данном случае это приводит к определенной ошибке) получим:

$$u_1 = 0\,, \qquad u_2 = u_1 + [\sigma_1 + \sigma_2]\, l/(8E)\,, \quad u_3 = u_2 + [\sigma_2 + \sigma_3]\, l/(8E)$$

$$u_4 = u_3\,, \qquad u_5 = u_4 + [\sigma_4 + \sigma_5]\, l/(8E)\,, \quad u_6 = u_5 + [\sigma_5 + \sigma_6]\, l/(8E)$$

или в общем случае:

$u_1 = 0$, - в начале отрезка;

$u_i = u_{i-1} + [\sigma_{i-1} + \sigma_i]\, h/(2E)$, i от (j -1)·k +2 до j·k с шагом 1- внутри j-го участка; (6)

$u_{j\cdot k+1} = u_{j\cdot k}$, j = m, m-1, ... ,1 - при переходе через j-й стык . (7)

Для обобщения формулы (7) на последний участок необходимо увеличить длину массива u до m+1.

После того, как выписаны все необходимые формулы, можно предложить следующий алгоритм вычислений:

- Описание массивов $X(6)$, $N(7)$, $\sigma(6)$, $u(7)$, $P(6)$.
- Описание функций $F(x)$, $q(x)$.
- Задание исходных данных (массивов X , P, значений k , m , l).
- Вычисление элементов массива N :σ

    внешний цикл по j от m до 1 с шагом -1

    $N_{j\cdot k} = N_{j\cdot k+1} + P_j$

    внутренний цикл по i от j·k -1 до (j - 1)·k +1 с шагом -1

    $N_i = N_{i+1} + [q(x_{i+1}) + q(x_i)]\, h/2$

    конец внутреннего цикла

    конец внешнего цикла.
- Вычисление элементов массива σ по формулам (2) :

    цикл по i от 1 до m·k с шагом 1
- Вычисление элементов массива u :

    $u_1 = 0$

    внешний цикл по j от 1 до m с шагом 1

    внутренний цикл по i от (j -1)·k +2 до j·k с шагом 1

    $u_i = u_{i-1} + [\sigma_{i-1} + \sigma_i]\, h/(2E)$

    конец внутреннего цикла

    $u_{j\cdot k+1} = u_{j\cdot k}$

    конец внешнего цикла.
- Печать значений x_i , N_i , σ_i , u_i :

    цикл по i от 1 до 6 с шагом 1

В качестве альтернативного можно предложить следующий вариант решения задачи. Интеграл, входящий в формулу для усилий (1) , вычислить аналитически. Это позволяет выражение для вычислений напряжений оформить в программе как функцию. Перемещения при этом можно вычислять не по формуле трапеций по трем точкам на участке (как в предыдущей программе), а используя более точный метод (например, стандартную программу метода Гаусса QG8).

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра прочности летательных аппаратов

Паспорт зачета

по дисциплине «Практикум по алгоритмизации и программированию», 3 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется из двух вопросов. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет ФЛА

Билет № _____
к зачету по дисциплине «Практикум по алгоритмизации и программированию»

---

1. Метод наименьших квадратов (МНК)
2. Программирование итерационного процесса в MathCAD .

Утверждаю: зав. кафедрой ____ _____________________ должность, ФИО
(подпись)
(дата)

2. Критерии оценки

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен описать алгоритмы или составить простейшие программы, оценка составляет менее 0,25 максимального балла БРС.
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы с трудом дает определение основных понятий, может составить простейшую программу, оценка составляет до 0,5 максимального балла БРС.
- Ответ на билет для зачета билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные алгоритмы, может их реализовать в программный код, оценка составляет до 0,75 максимального балла БРС.
- Ответ на билет для зачета билет засчитывается на **продвинутом** уровне, если студент

при ответе на вопросы проводит сравнительный анализ алгоритмов, без труда переводит их в программный код, оценка составляет более 0,75 максимального балла БРС.

3. Шкала оценки

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Вопросы к** зачету **по дисциплине** «Практикум по алгоритмизации и программированию» **(семестр 3)**

1. Понятие аппроксимации.
2. Аппроксимация в Excel с использованием встроенной функции.
3. Линия тренда.
4. Метод наименьших квадратов (МНК).
5. Реализация МНК в Excel.
6. Реализация МНК в MathCAD.
7. Реализация МНК в Fortran.
8. Алгоритм построения равномерной сетки.
9. Алгоритм построения сетки для разрывной функции.
10. Особенности работы с мастером диаграмм в Excel.
11. Особенности работы с мастером диаграмм в MathCAD.
12. Интегрирование табличных функций в Excel.
13. Интегрирование табличных функций в MathCAD.
14. Понятие собственных значений и векторов.
15. Метод простых итераций.
16. Программирование итерационного процесса в Excel.
17. Программирование итерационного процесса в MathCAD .
18. Программирование итерационного процесса в Fortran.
19. Встроенные процедуры на собственные значения в MathCAD.
20. Графическое решение нелинейных уравнений.
21. Алгоритм отделения корней нелинейных уравнений.
22. Алгоритмы уточнения корней.
23. Встроенные процедуры решения нелинейных уравнений.
24. Представление комплексных величин в прикладных пакетах.
25. Использование комплексных функций.
26. Программирование области при комфортном отображении.

ОПРЕДЕЛЕНИЕ СОБСТВЕННЫХ ЧАСТОТ КОЛЕБАНИЙ СИСТЕМЫ С НЕСКОЛЬКИМИ СТЕПЕНЯМИ СВОБОДЫ

ЗАДАНИЕ для пружинно-массовой механической системы методом простых итераций вычислить высшую и низшую частоты и соответствующие формы колебаний системы.

ПОСТАНОВКА ЗАДАЧИ Пусть имеется механическая система, состоящая из n масс m_j ($j = 1, 2, \dots ,n$), соединенных между собой и основанием пружинами жесткостью c_k . На рисунке 1 показана одна из таких систем для $n = 3$. Массы могут двигаться только вертикально, без поворотов. В этом случае число степеней свободы и, соответственно, число обобщенных координат, необходимых для описания положения системы равно n . Выберем в качестве обобщенных координат смещения x_j ($j = 1, 2, \dots ,n$) сосредоточенных масс m_j от положения равновесия.

Уравнения свободных колебаний системы около положения равновесия

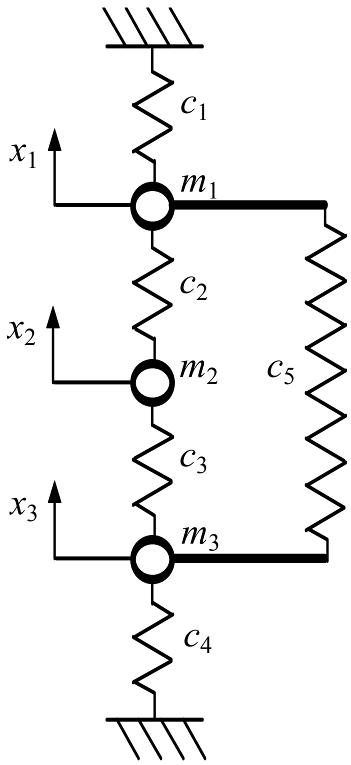


Рис. 1.

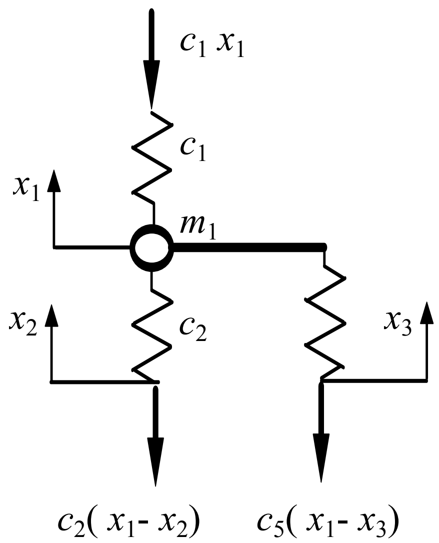


Рис. 2.

можно получить, используя II-й закон Ньютона.

Запишем уравнения II-й закона Ньютона для каждой массы

$$m_j \ddot{x}_j = F_j, \qquad (j = 1,2,\dots,n), \qquad \ddot{x}_j = \frac{d^2 x_j}{d\ t^2}.$$

Восстанавливающие силы F_j , вычисляются как суммы сил, действующие на j-ю массу со стороны пружин, соединенных с ней, при отклонении массы от положения равновесия,. Усилие в каждой пружине численно равно произведению жесткости пружины (c_j) на ее удлинение (разности смещений концов пружины).

Так, на массу m_1 системы (рис. 1) действуют три силы (рис. 2): со стороны верхней пружины - $c_i\ x_1$ (знак «минус» показывает, что сила действует в противоположную сторону от направления x_1), со стороны левой нижней пружины - $c_2\ (x_1 - x_2)$ (здесь $(x_1 - x_2)$ - удлинение пружины), со стороны правой нижней пружины - $c_5\ (x_1 - x_3)$. Тогда уравнение движения для m_1 имеет вид

$$m_1\ddot{x}_1 = -c_1x_1 - c_2(x_1 - x_2) - c_5(x_1 - x_3)$$

Аналогично можно получить уравнения движения для всех масс. В итоге имеем систему уравнений движения

$$\begin{aligned} m_1\ddot{x}_1 &= -c_1x_1 - c_2(x_1 - x_2) - c_5(x_1 - x_3) \\ m_2\ddot{x}_2 &= -c_2(x_2 - x_1) - c_3(x_2 - x_3) \\ m_3\ddot{x}_3 &= -c_4x_3 - c_3(x_3 - x_2) - c_5(x_3 - x_1) \end{aligned} \tag{1}$$

Решение системы дифференциальных уравнений (1) разыскивается в виде

$$x_k = X_k e^{i\omega t}, \qquad (k = 1, 2, 3), \tag{2}$$

где: X_k -амплитуда колебаний k-ой массы, ω - круговая частота.
Подставляя функции (2) в уравнения (1) , и сокращая на $e^{i\omega t}$, получим систему линейных однородных алгебраических уравнений

$$\begin{aligned} m_1\omega^2 X_1 &= c_1X_1 + c_2(X_1 - X_2) + c_5(X_1 - X_3) \\ m_2\omega^2 X_2 &= c_2(X_2 - X_1) + c_3(X_2 - X_3) \\ m_3\omega^2 X_3 &= c_4X_3 + c_3(X_3 - X_2) + c_5(X_3 - X_1) \end{aligned} \tag{2}$$

Вводя обозначения $\lambda = c_1 / m_1\omega^2$, представим систему (1) следующим образом

$$X_1 = \lambda\,[(1 + \frac{c_2}{c_1} + \frac{c_5}{c_1})X_1 - \frac{c_2}{c_1}X_2 - \frac{c_5}{c_1}X_3]$$

$$X_2 = \lambda\,[-\frac{c_2}{c_1}\frac{m_1}{m_2}X_1 + (\frac{c_2}{c_1} + \frac{c_3}{c_1})\frac{m_1}{m_2}X_2 - \frac{c_3}{c_1}\frac{m_1}{m_2}X_3]$$

$$X_3 = \lambda\,[-\frac{c_5}{c_1}\frac{m_1}{m_3}X_1 - \frac{c_3}{c_1}\frac{m_1}{m_3}X_2 + (\frac{c_3}{c_1} + \frac{c_4}{c_1} + \frac{c_5}{c_1})\frac{m_1}{m_3}X_3]$$

или в матричном виде

$$\{X\} = \lambda\,[H]\{X\} \tag{3}$$

где $\{X\}^{\mathrm{T}} = \{X_1, X_2, X_3\}$, а элементы матрицы $[H]$ вычисляются по формулам:

$$H_{11} = 1 + \frac{c_2}{c_1} + \frac{c_5}{c_1}, \quad H_{12} = -\frac{c_2}{c_1}, \quad H_{13} = -\frac{c_5}{c_1},$$

$$H_{21} = -\frac{c_2}{c_1}\frac{m_1}{m_2}, \quad H_{22} = (\frac{c_2}{c_1} + \frac{c_3}{c_1})\frac{m_1}{m_2}, \quad \text{и т.д.}$$

Для определения минимального собственного значения λ и соответствующего ему собственного вектора $\{ X \}$ системы (3) можно воспользоваться несколькими методами, в частности, методом простых итераций, который заключается в следующем.

Выбирается некоторое начальное приближение для собственного вектора

$$\{ X \}^{(0)\,T} = \{ X_1^{(0)}, X_2^{(0)}, X_3^{(0)} \}$$

по формуле

$$\{ X \}^{(1)} = [\, H \,]\, \{ X \}^{(0)}$$

или в общем случае

$$\{ X \}^{(i+1)} = [\, H \,]\, \{ X \}^{(i)}$$

На каждом шаге итераций определяются $\lambda_k^{(i+1)} = X_k^{(i)} / X_k^{(i+1)}$, $k = 1, 2, \ldots, n$,
С ростом числа итераций $\lambda_k^{(i+1}$ сходятся к минимальному собственному значению λ системы (3) . При достижении определенной точности в вычислении λ процесс итерации прекращается.

В процессе итераций абсолютные значения компонент вектора $\{ X \}$ растут, поэтому при численных расчетах после каждого шага итераций необходимо выполнять нормирование $\{ X \}$, например, по значению максимального модуля его элементов.

ЗАДАНИЕ. Для своего варианта

1. Получить систему дифференциальных уравнений движения в виде (1).
2. Записать систему однородных алгебраических уравнений в виде (2), (3). Выписать выражения для определения элементов матрицы $[\, H \,]$.
3. Составить подпрограмму, в которой методом простых итераций вычисляются минимальные значения λ и соответствующий собственный вектор $\{ X \}$ для системы (3).
4. Составить головную программу для вычислений min λ (для матрицы $[\, H \,]$) и max λ (для матрицы $[\, H \,]^{-1}$).

АЛГОРИТМ РЕШЕНИЯ

1. Составить подпрограмму. в которой вычисляются компоненты матрицы $[\, H \,]$.
2. Составить головную программу, в которой:
   - вводятся исходные данные ($\frac{c_2}{c_1}$, $\frac{c_3}{c_1}$, … , $\frac{m_1}{m_2}$, …)
   - вызывается подпрограмма вычисления $[\, H \,]$
   - вызывается подпрограмма метода простых итераций
   - выводятся результаты (λ , $\{ X \}$, число итераций)

Данные из головной программы в подпрограмму вычисления $[\, H \,]$ можно передавать либо через список формальных параметров, либо через общие области, используя оператор COMMON.

Подпрограмму метода простых итераций можно организовать следующим образом.

<u>Формальные параметры</u> H, X, Y, Z, AL, N, E ,m, mmax, IER где:

входные параметры: H (N, N) - матрица системы (3), N - порядок системы, E - точность вычислений; mmax - максимальное число итераций;
выходные параметры: AL - значение λ_{min} системы (3), X(N) - соответствующий λ_{min} собственный вектор; m - число выполненных итераций; IER - код ошибки (IER = 0 - процесс сошелся, в противном случае IER=1).
массивы: Y(N), Z(N) - вспомогательные массивы для хранения результатов промежуточных вычислений.

Алгоритм подпрограммы

a) описание используемых массивов
b) задание начальных значений X
c) вычисление последующих приближений Y=H·X
d) вычисление значений $Z_i = X_i / Y_i$, i = 1,..., n
e) найти среднеквадратичную ошибку

$$E1 = \left| \frac{1}{|z_n|} \frac{1}{n-1} [\sum_{i=1}^{n-1} (z_i - z_n)^2]^{1/2} \right|$$

f) найти максимальный модуль элемента Y
   $S = \max(| Y_1|, | Y_2|, \dots ,| Y_n|)$
g) вычислить начальное приближение для следующего шага итераций
   $X_i = Y_i / S$, i = 1, n
h) выполнить проверку на прекращение итераций:
   если $E1 - E > 0$, перейти на пункт c)
i) выполнить проверку на прекращение итераций по mmax:
   если m= mmax , выйти из подпрограммы с кодом ошибки IER=1
j) запомнить вычисленное значение λ (последнее значение Z_1)
   $AL = Z_1$
k) вернуться в вызывающую программу.

ВАРИАНТЫ ЗАДАНИЯ. Исходная расчетная схема представляет собой цепочку из 5-ти масс и 6-ти жесткостей. В зависимости от варианта, одна из пружин может иметь нулевую жесткость (не учитывается при выводе уравнений движения),а некоторые массы могут быть соединены между собой или с основанием дополнительной "параллельной" жесткостью. В таблице 1 приведены номер нулевой жесткости и точки, соединенные "параллельной" жесткостью. В таблице 2 в зависимости от последней цифры кода группы заданны значения отношений масс и нулевых жесткостей.

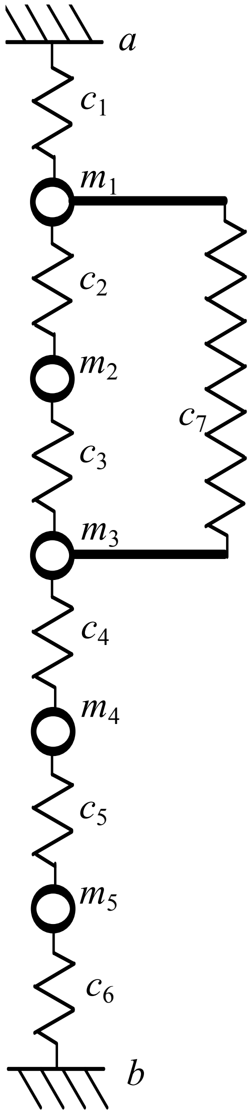


Рис. 3.

Таблица 1. Варианты расчетных схем.

№ вар. 1	Нулевые жесткости 2	Парал-ые жесткости 4	№ вар. 1	Нулевые жесткости 2	Парал-ые жесткости 4
1	1	—	16	—	4-b
2	1	1-3	17	—	a-3
3	1	2-4	18	—	1-4
4	1	3-5	19	—	2-5
5	1	4-b	20	—	1-5
6	1	1-4	21	2	1-3
7	1	2-5	22	3	2-4
8	1	3-b	23	4	3-5
9	1	1-5	24	3	1-4
10	1	2-b	25	3	1-5
11	—	—	26	1	2-4
12	—	1-3	27	1	1-3
13	—	2-4	28	1	2-4
14	—	3-5	29	1	2-5
15	—	a-2	30	1	2-5

Таблица 2. Варианты жесткостей и масс.

№	Жесткости c_i / c i						Массы m_i / m				
вар.	i=1	i=2	i=3	i=4	i=5	i=6	i=1	i=2	i=3	i=4	i=5
1	1.0	2.0	0.8	1.2	0.9	1.5	1.0	0.5	1.2	2.0	0.8
2	1.0	1.5	0.5	2.0	2.0	1.0	1.0	1.0	2.0	3.0	0.5

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра прочности летательных аппаратов

Паспорт
расчетно-графического задания (работы)

по дисциплине «Практикум по алгоритмизации и программированию», 3 семестр

1. **Методика оценки**

В семестре студенты выполняют два расчетно-графического задания (работы) (РГЗ)(Р).

В рамках первой РГЗ студенты должны рассчитать на растяжение-сжатие прямой брус переменного сечения.

При выполнении расчетно-графического задания (работы) студенты должны составить алгоритмы и программы вычисления усилии, напряжений и перемещений сечений бруса.

Обязательные структурные части РГЗ.

- Определение варианта расчета по двоичному коду.
- Изображение расчетной схемы в масштабе.
- Разработка алгоритма построения сетки для разрывной функции.
- Вычисление значений функции по участкам.
- Численное интегрирование табличной функции.

В рамках второго РГЗ студенты должны вычислить из нелинейного уравнения в виде равенства нулю определителя системы однородных СЛАУ собственные значения (частоты колебаний) и построить собственные функции в задаче о колебаниях стержня.
Обязательные структурные части РГЗ.

- Составление определителя системы по варианту задания
- Составления алгоритма и программы поиска корней нелинейного уравнения
- Нормирование собственных функций
- Построение форм колебаний

2. **Критерии оценки**

- Работа считается **не выполненной,** если выполнены не все части РГЗ(Р), отсутствует описание алгоритмов и программ, допущены грубые ошибки в результатах, оценка составляет менее 0,25 максимального балла, указанного в БРС.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ(Р) выполнены формально: не приведено описание алгоритмов, допущены неточности в результатах, оценка составляет более 0,25 максимального балла, указанного в БРС.
- Работа считается выполненной **на базовом** уровне, если содержит все обязательные структурные части, не содержит неточностей, оценка составляет менее 0,75 максимального балла, указанного в БРС.
- Работа считается выполненной **на продвинутом** уровне при качественном оформлении и безошибочном выполнении всех структурных частей РГР, оценка составляет более 0.75 балла, указанного в БРС.

3. **Шкала оценки**

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Задание для РГЗ(Р) №1**

ЦЕНТРАЛЬНОЕ РАСТЯЖЕНИЕ И СЖАТИЕ ПРЯМОГО БРУСА.

ЗАДАНИЕ: вычислить таблицы значений и построить эпюры продольных сил N, нормальных напряжений σ и перемещений u поперечных сечений прямого бруса (рис. 1).

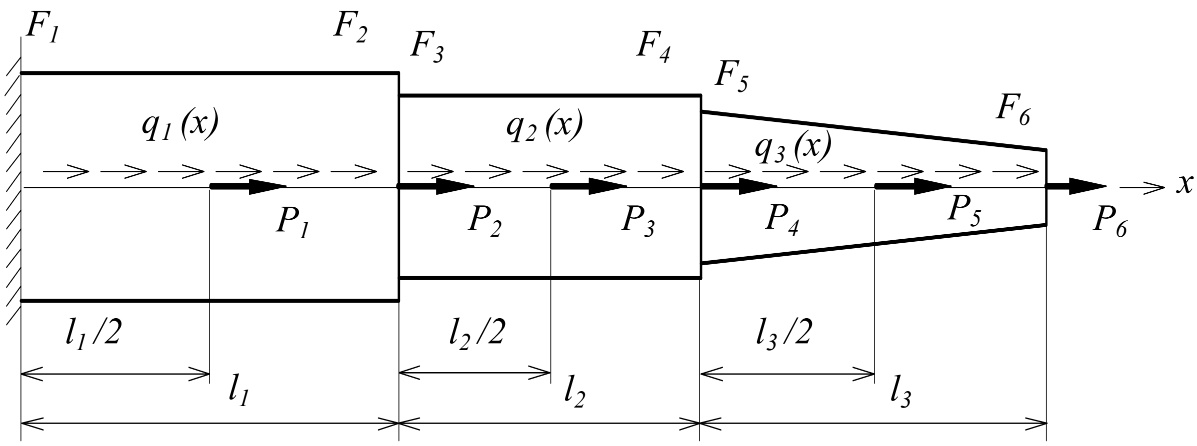


Рис. 1

Геометрически брус можно разделить на три участка. На i-ом участке (i =1, 2, 3) длиной l_i заданы F_{2i+1} , F_{2i} - площади левого и правого сечений участка бруса (на каждом участке принят линейный закон измененения площади по длине); P_{2i+1} , P_{2i} - сосредоточенные силы, приложенные в соответствии со схемой (рис. 1); q_i (x) - продольная распределенная нагрузка, изменяющаяся по линейному закону (заданы k_i - тангенс наклона в уравнении прямой и b_i - значение q_i на левой границе участка).

Для вычисления значений N, σ, u необходимо воспользоваться следующими соотношениями:

$$N(x) = \int_{x}^{l_1+l_2+l_3} q(x)dx + \sum P_j \qquad (1)$$

В сумме (1) учитываются все сосредоточенные силы, приложенные на участке от x до l_1+l_2+l_3 .

$$\sigma(x) = N(x) / F(x) \qquad (2)$$

$$u\ (x) = \frac{1}{E}\int_{0}^{x} \sigma\,(x)d\,x \qquad (3)$$

Для решения задачи требуется разбить брус на m силовых участков (в общем случае m = 6) и на каждом участке силовом вычислить требуемые величины. Решение представить в виде таблицы и эпюр. На каждом силовом участке рассчитать не менее трех точек.

Для определения варианта задания порядковый номер по групповому журналу перевести в двоичный код и дополнить нулями до 6-ти позиций (например №3=000011). В зависимости от значений в каждой позиции (0 или 1) по таблице определить исходные данные. (E=1.1·10^{8} кН / м^{2})

№ позиции	1-ая			2-ая			3-ья		
значения	l_1	$F_1 \cdot 10^4$	$F_2 \cdot 10^4$	l_2	$F_3 \cdot 10^4$	$F_4 \cdot 10^4$	l_3	$F_5 \cdot 10^4$	$F_6 \cdot 10^4$
в позиции	м	м^2	м^2	м	м^2	м^2	м	м^2	м^2
0	0.5	5	3	0.4	4	3	0.5	3	3
1	0.3	4	4	0.7	5	3	0.4	4	2

№ позиции	4-ая				5-ая				6-ая			
значения	P_1	P_2	k_1	b_1	P_3	P_4	k_2	b_2	P_5	P_6	k_3	b_3
в позиции	кН	кН	кН/м^2	кН/м	кН	кН	кН/м^2	кН/м	кН	кН	кН/м^2	кН/м
0	0	10	5	2	0	-5	-4	-4	0	20	-5	-3
1	10	-10	0	0	-10	20	0	5	15	-5	0	0

При решении задачи возможны многие варианты программ. Рассмотрим на примере упрощенной схемы бруса (рис. 2) алгоритм одного из них.

ПОСТАНОВКА ЗАДАЧИ

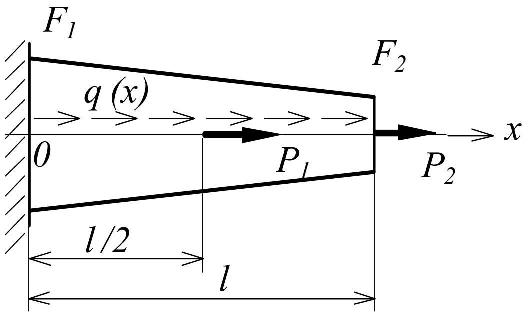


Рис. 2

Исходные данные: l = 1м , $F_1 = 4 \cdot 10^{-4}$ м ^{2}, $F_2 = 2 \cdot 10^{-4}$ м ^{2}, P_1 = 10 Кн , P_2 = -20 Кн , k = 5 Кн / м^2, b = -5 Кн / м .

Уравнение изменения площади поперечного сечения бруса F(x) запишется в виде

$$F(x) = -2 \cdot 10^{-4} x + 4 \cdot 10^{-4}$$

Закон изменения распределенной нагрузки

$$q(x) = 5x - 5$$

Так как при $x = l/2$ в точке приложения силы P_1 продольная сила N терпит разрыв, брус необходимо разбить на два силовых участка: 1-ый участок $0 < x < l/2$; 2-ой участок $l/2 < x < l$. При этом в точке $x = l/2$ величины N и σ необходимо вычислить дважды: слева и справа от точки.

АЛГОРИТМ РЕШЕНИЯ.

Введем в рассмотрение массив $X(6)$, элементы которого числено равны координатам точек в которых вычисляются N , σ , u . При трех точках на каждом участке и равномерной сетке получим:

$$x_1 = 0,\quad x_2 = l/4,\quad x_3 = l/2,\quad x_4 = l/2,\quad x_5 = 3l/4,\quad x_6 = l .$$

Для вычислений и хранения усилий, напряжений и перемещений в точках x_i , будем использовать массивы $N(6)$, $\sigma(6)$, $u(6)$.

Вычисление усилий методом сечений удобно начинать со свободного конца бруса. Из расчетной схемы видно, что

$$N_6 = P_2 .$$

Для N_5 из (1) имеем:

$$N_5 = N_6 + \int_{x_5}^{x_6} q(x)dx$$

Вычисляя последний интеграл по формулам трапеций с использованием значений функции $q(x)$ только на концах отрезка интегрирования (т. к. $q(x)$ - линейная функция, то результат будет точным), получим

$$N_5 = N_6 + [q(x_6) + q(x_5)]\, h/2$$

где $h = x_6 - x_5$ - шаг интегрирования. В рассматриваемом случае для равномерности сетки независимо от пределов интегрирования шаг постоянен и равен $h = l/4$.

Для N_4 аналогично получим формулу

$$N_4 = N_5 + [q(x_5) + q(x_4)]\, l/8$$

В точке разрыва имеем: $N_3 = N_4 + P_1$

и далее: $N_2 = N_3 + [q(x_3) + q(x_2)]\, l/8$

$$N_1 = N_2 + [q(x_2) + q(x_1)]\, l/8$$

Если увеличить длину массива N на единицу и ввести значение $N_7 = 0$, то вместо $N_6 = P_2$ получим $N_6 = N_7 + P_2$.

Тогда можно независимо от количества силовых участков и точек разбиения внутри участков записать следующие рекуррентные формулы для определения усилий в поперечных сечениях бруса:

$N_{m \cdot k+1} = 0$ - значение усилий за пределом бруса;

$N_{j \cdot k} = N_{j \cdot k+1} + P_j$, j = m, m-1, ... ,1 - при переходе через j-ый стык (стык j-го и j+1 -го силовых участков);

$N_i = N_{i+1} + [q(x_{i+1}) + q(x_i)]\, h/2$, $i = j \cdot k + 1, \ldots, (j-1) \cdot k + 1$ - для точек внутри j-го силового участка.

Здесь: m - число участков; k - количество точек на участке; h - шаг разбиения. Если при разбиении используется неравномерная сетка, то $h_i = x_{i+1} - x_i$.

Зная усилия и напряжения (последние определяются в соответствии с формулой (2)), можно вычислить перемещения в контрольных точках по формуле (3). При интегрировании по формуле трапеций (в данном случае это приводит к определенной ошибке) получим:

$$u_1 = 0, \qquad u_2 = u_1 + [\sigma_1 + \sigma_2]\, l/(8E), \quad u_3 = u_2 + [\sigma_2 + \sigma_3]\, l/(8E)$$

$$u_4 = u_3, \qquad u_5 = u_4 + [\sigma_4 + \sigma_5]\, l/(8E), \quad u_6 = u_5 + [\sigma_5 + \sigma_6]\, l/(8E)$$

или в общем случае:

$u_1 = 0$, - в начале отрезка;

$u_i = u_{i-1} + [\sigma_{i-1} + \sigma_i]\, h/(2E)$, i от $(j-1) \cdot k + 2$ до $j \cdot k$ с шагом 1- внутри j-го участка; (6)

$u_{j \cdot k+1} = u_{j \cdot k}$, j = m, m-1, ... ,1 - при переходе через j-й стык. (7)

Для обобщения формулы (7) на последний участок необходимо увеличить длину массива u до m+1.

После того, как выписаны все необходимые формулы, можно предложить следующий алгоритм вычислений:

- Описание массивов $X(6)$, $N(7)$, $\sigma(6)$, $u(7)$, $P(6)$.
- Описание функций $F(x)$, $q(x)$.
- Задание исходных данных (массивов X, P, значений k, m, l).
- Вычисление элементов массива N :σ

  внешний цикл по j от m до 1 с шагом -1

  $N_{j \cdot k} = N_{j \cdot k+1} + P_j$

  внутренний цикл по i от $j \cdot k + 1$ до $(j-1) \cdot k + 1$ с шагом -1

  $N_i = N_{i+1} + [q(x_{i+1}) + q(x_i)]\, h/2$

  конец внутреннего цикла

  конец внешнего цикла.
- Вычисление элементов массива σ по формулам (2) :

цикл по i от 1 до m·k с шагом 1

- Вычисление элементов массива u :

$u_1 = 0$

внешний цикл по j от 1 до m с шагом 1

внутренний цикл по i от (j -1)·k +2 до j·k с шагом 1

$u_i = u_{i-1} + [\sigma_{i-1} + \sigma_i]\, h/(2E)$

конец внутреннего цикла

$u_{j \cdot k+1} = u_{j \cdot k}$

конец внешнего цикла.

- Печать значений x_i, N_i, σ_i, u_i :

цикл по i от 1 до 6 с шагом 1

В качестве альтернативного можно предложить следующий вариант решения задачи. Интеграл, входящий в формулу для усилий (1) , вычислить аналитически. Это позволяет выражение для вычислений напряжений оформить в программе как функцию. Перемещения при этом можно вычислять не по формуле трапеций по трем точкам на участке (как в предыдущей программе), а используя более точный метод (например, стандартную программу метода Гаусса QG8).

5. **Задание для РГЗ(Р) №2**

ОПРЕДЕЛЕНИЕ СОБСТВЕННЫХ ФОРМ КОЛЕБАНИЙ УПРУГОЙ БАЛКИ.

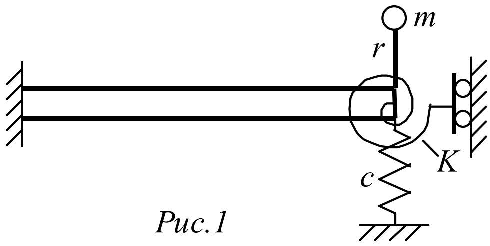


Рис.1

Обозначения:

Y(x) - прогиб оси балки;

EJ - жесткость балки на изгиб;

l - длина балки;

ρF - погонная масса;

m - сосредоточенная масса;

r - длина жесткого рычага - расстояние между осью балки и массой *m*;

c - линейная жесткость упругой опоры;

K - угловая жесткость упругой опоры;

$z = x/l$ - безразмерная координата ($0 \le z \le 1$).

При решении задач на определение собственных форм колебаний упругих балок функцию прогиба удобно записывать через функции Крылова S(x), T(x), U(x), V(x), которые представляют собой линейные комбинации гиперболических и тригонометрических синусов и косинусов:

$$S(x) = [\mathrm{ch}(x) + \cos(x)]/2, \quad T(x) = [\mathrm{sh}(x) + \sin(x)]/2$$
$$U(x) = [\mathrm{ch}(x) - \cos(x)]/2, \quad V(x) = [\mathrm{sh}(x) - \sin(x)]/2. \qquad (1)$$

Собственные формы $Y(z)$ колебаний консольной балки (рис.1) через функции Крылова определяются как

$$Y(z) = U(\alpha z) + D V(\alpha z) \qquad (2)$$

Параметры α и D зависят от граничных условий на правом конце балки. Собственные значения α (безразмерные собственные частоты) определяются из следующего характеристического уравнения:

$$\begin{vmatrix} \mathrm{V}(\alpha)+\left(f\alpha-\dfrac{g}{\alpha^3}\right)\mathrm{U}(\alpha) & \mathrm{S}(\alpha)+\left(f\alpha-\dfrac{g}{\alpha^3}\right)\mathrm{V}(\alpha) \\ \mathrm{S}(\alpha)-\left(w\alpha^3-\dfrac{q}{\alpha}\right)\mathrm{T}(\alpha) & \mathrm{T}(\alpha)-\left(w\alpha^3-\dfrac{q}{\alpha}\right)\mathrm{U}(\alpha) \end{vmatrix}=0 \qquad (3)$$

где f,g,w,q-безразмерные параметры: $f = m/\rho Fl$, $g = cl^3/EJ$, $w = f(r/l)^2$, $q = Kl/EJ$.

Уравнение (3) имеет бесчисленное множество положительных корней α , i =1,2,3, ... Для каждого значения а коэффициент D определяется из уравнения

$$\mathrm{V}(\alpha)+\left(f\alpha-\frac{g}{\alpha^3}\right)\mathrm{U}(\alpha)+D\left[\mathrm{S}(\alpha)+\left(f\alpha-\frac{g}{\alpha^3}\right)\mathrm{V}(\alpha)\right]=0 \qquad (4)$$

Каждой паре значений α и D соответствует своя собственная форма Y(z), которая определяется с точностью до множителя. Собственные формы можно пронормировать, например, так, чтобы максимальное значение нормированной формы было равно единице. Тогда для нормированной формы можно записать следующее выражения

$$Y(z) = Y(z)/B = [\mathrm{U}(\alpha z) + D\mathrm{V}(\alpha z)]\,/\,B$$

$$B = \max|Y(z)| \ (6) \qquad (5)$$

Точкой α , в которой функция $Y(z)$ принимает максимальное по модулю значение B, может быть либо экстремум функции, либо точка z =1. В первом случае для определения z необходимо найти корни уравнения $Y'(z)$=0, или в развернутом виде

$$\mathrm{T}(\alpha z) + D\mathrm{U}(\alpha z) = 0 \qquad (7)$$

Так как корней на отрезке 0 < z < 1 может быть несколько (в вариантах задания - не более трех), то необходимо найти все. Затем из найденных точек и точки z = 1 выбрать в качестве z ту, в которой $|Y(z)|$ принимает максимальное значение.

Можно использовать и другой способ нормирования, при котором коэффициент В вычисляется по формуле

$$B=\int_0^1 [Y(z)]^2\,dz \qquad (8)$$

Если собственная форма имеет узлы (точки, где функция $Y(z)$ меняет знак), то интеграл от 0 до 1 надо вычислять как сумму интегралов по участкам между узлами. Поэтому сначала необходимо найти эти узлы, т.е. корни уравнения (2). В зависимости от варианта задания на отрезке от 0 до 1 их может быть не более двух.

ЗАДАНИЕ. Для своего варианта исходных данных вычислить три первых собственных значения α из уравнения (3) и построить графики соответствующих собственных нормированных форм.

ВЫБОР ВАРИАНТА ЗАДАНИЯ. Переведите номер по групповому журналу из десятичной в двоичную систему исчисления. Запишите перед полученным числом нули так, чтобы получился пятизначный код. По двоичным значениям в каждой позиции кода определяются исходные данные (до 23-го варианта):

N позиции (слева направо)	1	2	3	4	5
Параметр	f	w	g	q	способ нормир.
0 в позиции	1	0	0	0	(6)
1 в позиции	0	1	1	1	(8)

РЕКОМЕНДАЦИИ к составлению программы. Для решения нелинейных уравнений можно использовать подпрограммы отделения корней и метода половинного деления (см. задачу 5), для численного интегрирования - подпрограмму метода Гаусса (см. задачу 6). Необходимо также составить подпрограммы-функции и (или) подпрограммы для вычисления: функций Крылова; функций (2),(3),(5),(7); подынтегральной функции в формуле (8); коэффициента B по формулам (6) либо (8).

В головной программе требуется описать массивы:

- A(3) -= для собственных значений α;
- D(3) - для коэффициентов D;
- Z0(3) - для точек экстремумов (или узлов);
- Y0(3) – для значений функции Y в точках экстремумов;
- B(3) - для коэффициентов B в формулах (6) или (8).

Алгоритм может быть построен по следующей схеме. В цикле по номеру формы (i=1,2,3) вычисляются:

- A(i) - собственное значение α;
- D(i) - коэффициент D;
- k значений Z0 и Y0 - точки экстремумов i-ой формы и значения функции Y в этих точках ($k \le 3$ - зависит от номера формы) или k значений координат узлов i-ой формы ($k \le 2$);
- B(i) - коэффициент B по формуле (6) или (8). Все результаты выводятся на печать.

После завершения цикла по i вычисляются при изменении z от 0 до 1 с шагом не более 0.1 и печатаются в таблицу ординаты трех собственных форм. При этом для записи и хранения текущих значений можно использовать массив Z0 или Y0.

18.ОПРЕДЕЛЕНИЕ СОБСТВЕННЫХ ФОРМ КОЛЕБАНИЙ УПРУГОЙ БАЛКИ.

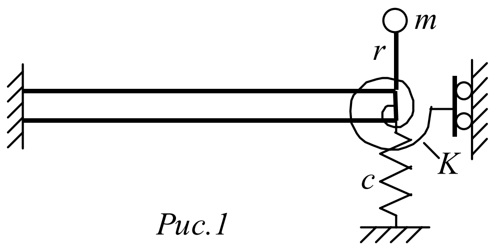


Рис.1

Обозначения:

Y(x) - прогиб оси балки;
EJ - жесткость балки на изгиб;
l - длина балки;
ρF - погонная масса;
m - сосредоточенная масса;
r - длина жесткого рычага - расстояние между осью балки и массой *m*;
c - линейная жесткость упругой опоры;
K - угловая жесткость упругой опоры;
$z = x/l$ - безразмерная координата ($0 \le z \le 1$).

При решении задач на определение собственных форм колебаний упругих балок функцию прогиба удобно записывать через функции Крылова S(x), T(x), U(x), V(x), которые представляют собой линейные комбинации гиперболических и тригонометрических синусов и косинусов:

$$\mathrm{S}(x) = [\mathrm{ch}(x) + \cos(x)]\,/\,2, \qquad \mathrm{T}(x) = [\mathrm{sh}(x) + \sin(x)]\,/\,2$$
$$\mathrm{U}(x) = [\mathrm{ch}(x) - \cos(x)]\,/\,2, \qquad \mathrm{V}(x) = [\mathrm{sh}(x) - \sin(x)]\,/\,2. \tag{1}$$

Собственные формы $Y(z)$ колебаний консольной балки (рис.1) через функции Крылова определяются как

$$Y(z) = \mathrm{U}(\alpha z) + D\mathrm{V}(\alpha z) \tag{2}$$

Параметры α и D зависят от граничных условий на правом конце балки. Собственные значения α (безразмерные собственные частоты) определяются из следующего характеристического уравнения:

$$\begin{vmatrix} \mathrm{V}(\alpha)+\left(f\alpha-\dfrac{g}{\alpha^3}\right)\mathrm{U}(\alpha) & \mathrm{S}(\alpha)+\left(f\alpha-\dfrac{g}{\alpha^3}\right)\mathrm{V}(\alpha) \\ \mathrm{S}(\alpha)-\left(w\alpha^3-\dfrac{q}{\alpha}\right)\mathrm{T}(\alpha) & \mathrm{T}(\alpha)-\left(w\alpha^3-\dfrac{q}{\alpha}\right)\mathrm{U}(\alpha) \end{vmatrix} = 0 \tag{3}$$

где f,g,w,q-безразмерные параметры: $f = m/\rho Fl$, $g = cl^3/EJ$, $w = f(r/l)^2$, $q = Kl/EJ$.

Уравнение (3) имеет бесчисленное множество положительных корней α, i =1,2,3, ... Для каждого значения а коэффициент D определяется из уравнения

$$\mathrm{V}(\alpha)+\left(f\alpha-\frac{g}{\alpha^3}\right)\mathrm{U}(\alpha)+D\left[\mathrm{S}(\alpha)+\left(f\alpha-\frac{g}{\alpha^3}\right)\mathrm{V}(\alpha)\right]=0 \tag{4}$$

Каждой паре значений α и D соответствует своя собственная форма Y(z), которая определяется с точностью до множителя. Собственные формы можно пронормировать, например, так, чтобы максимальное значение нормированной формы было равно единице. Тогда для нормированной формы можно записать следующее выражения

$$Y(z) = Y(z)/B = [\mathrm{U}(\alpha z) + D\mathrm{V}(\alpha z)]\ /\ B \qquad (5)$$

$$B = \max|Y(z)| \quad (6)$$

Точкой α , в которой функция $Y(z)$ принимает максимальное по модулю значение B, может быть либо экстремум функции, либо точка z =1. В первом случае для определения z необходимо найти корни уравнения $Y'(z)$=0, или в развернутом виде

$$\mathrm{T}(\alpha z) + D\mathrm{U}(\alpha z) = 0 \qquad (7)$$

Так как корней на отрезке 0 < z < 1 может быть несколько (в вариантах задания - не более трех), то необходимо найти все. Затем из найденных точек и точки z = 1 выбрать в качестве z ту, в которой $|Y(z)|$ принимает максимальное значение.

Можно использовать и другой способ нормирования, при котором коэффициент В вычисляется по формуле

$$B = \int_0^1 [Y(z)]^2\, dz \qquad (8)$$

Если собственная форма имеет узлы (точки, где функция $Y(z)$ меняет знак), то интеграл от 0 до 1 надо вычислять как сумму интегралов по участкам между узлами. Поэтому сначала необходимо найти эти узлы, т.е. корни уравнения (2). В зависимости от варианта задания на отрезке от 0 до 1 их может быть не более двух.

ЗАДАНИЕ. Для своего варианта исходных данных вычислить три первых собственных значения α из уравнения (3) и построить графики соответствующих собственных нормированных форм.

ВЫБОР ВАРИАНТА ЗАДАНИЯ. Переведите номер по групповому журналу из десятичной в двоичную систему исчисления. Запишите перед полученным числом нули так, чтобы получился пятизначный код. По двоичным значениям в каждой позиции кода определяются исходные данные (до 23-го варианта):

N позиции (слева направо)	1	2	3	4	5
Параметр	f	w	g	q	способ нормир.
0 в позиции	1	0	0	0	(6)
1 в позиции	0	1	1	1	(8)

РЕКОМЕНДАЦИИ к составлению программы. Для решения нелинейных уравнений можно использовать подпрограммы отделения корней и метода половинного деления (см. задачу 5), для численного интегрирования - подпрограмму метода Гаусса (см. задачу 6). Необходимо также составить подпрограммы-функции и (или) подпрограммы для вычисления: функций Крылова; функций (2),(3),(5),(7); подынтегральной функции в формуле (8); коэффициента B по формулам (6) либо (8).

В головной программе требуется описать массивы:

- A(3) -= для собственных значений α;
- D(3) - для коэффициентов D;
- Z0(3) - для точек экстремумов (или узлов);

- Y0(3) – для значений функции *Y* в точках экстремумов;
- B(3) - для коэффициентов *B* в формулах (6) или (8).

Алгоритм может быть построен по следующей схеме. В цикле по номеру формы (i=1,2,3) вычисляются:

- A(i) - собственное значение α;
- D(i) - коэффициент *D*;
- k значений Z0 и Y0 - точки экстремумов i-ой формы и значения функции *Y* в этих точках ($k \le 3$ - зависит от номера формы) или k значений координат узлов i-ой формы ($k \le 2$);
- B(i) - коэффициент *B* по формуле (6) или (8). Все результаты выводятся на печать.

После завершения цикла по i вычисляются при изменении *z* от 0 до 1 с шагом не более 0.1 и печатаются в таблицу ординаты трех собственных форм. При этом для записи и хранения текущих значений можно использовать массив Z0 или Y0.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра прочности летательных аппаратов

Паспорт зачета

по дисциплине «Практикум по алгоритмизации и программированию», 4 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется из двух вопросов. В ходе зачета преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет ФЛА

Билет № _____
к зачету по дисциплине «Практикум по алгоритмизации и программированию»

---

1. Использование функции Хэвисайда в алгоритме вычисления прогиба.
2.Задание функций моментов в виде матричной функции.

Утверждаю: зав. кафедрой ____ ____________________ должность, ФИО
(подпись)
(дата)

2. Критерии оценки

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен описать алгоритмы или составить простейшие программы, оценка составляет менее 0,25 максимального балла БРС.
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы с трудом дает определение основных понятий, может составить простейшую программу, оценка составляет до 0,5 максимального балла БРС.
- Ответ на билет для зачета билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные алгоритмы, может их реализовать в программный код, оценка составляет до 0,75 максимального балла БРС.
- Ответ на билет для зачета билет засчитывается на **продвинутом** уровне, если студент

при ответе на вопросы проводит сравнительный анализ алгоритмов, без труда переводит их в программный код, оценка составляет более 0,75 максимального балла БРС.

3. Шкала оценки

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Вопросы к** зачету **по дисциплине** «Практикум по алгоритмизации и программированию»**, 4-й семестр**

1. Алгоритм построения сетки для разрывной функции.
2. Алгоритм вычисления разрывной функции.
3. Использование функции Хэвисайда в алгоритме вычисления прогиба.
4. Построение функции прогиба балки в Excel.
5. Построение функции прогиба балки в MathCAD.
6. Табулирование функции прогиба балки в Fortran.
7. Построение эпюр с использованием мастера диаграмм в Excel.
8. Особенности работы с мастером диаграмм в MathCAD.
9. Построение эпюр в MathCAD по внешним табличным данным.
10. Составление уравнений моментов по участкам рамы.
11. Задание уравнений моментов в виде матричной функции.
12. Вычисление определенных интегралов от произведения функций.
13. Формирование системы линейных алгебраических уравнений.
14. Процедуры решения СЛАУ.
15. Задание функций моментов в виде матричной функции.
16. Особенности использования матричных функций в
17. Составление процедур вычисления функций силовых факторов в MathCAD.
18. Составление процедур вычисления функций силовых факторов в Fortran.
19. Особенности интегрирования по участкам с использованием процедур в MathCAD.
20. Особенности интегрирования по участкам с использованием процедур в Fortran.
21. Алгоритмы обработки матрицы перемещений по силовым факторам.

ПРИМЕНЕНИЕ КОМПЛЕКСНЫХ ВЕЛИЧИН.

ВВЕДЕНИЕ. При выполнении расчетов с использованием методов ТФКП, теории упругости и т.д. в ФОРТРАН-программах могут быть использованы комплексные величины. Объектом комплексного типа в ФОРТРАНе могут быть константа, переменная, элемент массива, результат обращения к функции, значение выражения.

Данное комплексного типа представляется парой значений вещественного типа. Первая компонента пары представляет действительную часть, а вторая - мнимую часть комплексного данного. Комплексные константы определяются по форме записи - это пара вещественных констант, разделенных запятой и заключенных в скобки; первая вещественная константа задает действительную часть комплексного числа, вторая - мнимую.

Комплексные переменные, массивы, внешние функции должны быть описаны в операторе описания типа COMPLEX. Например, оператор COMPLEX ZZ, Z1(4,4) описывает комплексную величину ZZ (переменную либо функцию) и двумерный массив Z1 , состоящий из ·16 комплексных элементов.

В ФОРТРАНе имеются встроенные функции, предназначенные для работы с комплексными величинами (ниже в таблице Z - комплексная, а X и Y - вещественные величины):

Тип функции	Функция	Значение
комплексный	CMPLX(X,Y)	X+i·Y
----’’----	CONJG(Z)	X-i·Y
----’’----	CSIN(Z)	sin(Z)
----’’----	CCOS(Z)	cos(Z)
----’’----	CEXP(Z)	exp(Z)
----’’----	CSQRT(Z)	sqrt(|Z|)·exp(i·arg(Z)/2)
----’’----	CLOG(Z)	ln(|Z|) +i·arg(Z)
вещественный	REAL(Z)	Re(Z)
----’’----	AIMAG(Z)	Im(Z)
----’’----	CABS(Z)	$\|Z\|=sqrt(X^2+Y^2)$
----’’----	ATAN2(Y,X)	arg(Z)=arctg(Y/X)

CSQRT(Z) sqrt(|Z|)·exp(i·arg(Z)/2) $-\pi \le arg(Z) \le \pi$

CLOG(Z) вычисляет главное значение (k=0) функции
$ln(Z) = ln(|Z|)+i\cdot(arg(Z)+2\cdot k\cdot\pi)$
Для вычисления других комплексных функций можно воспользоваться формулами:

$tg(Z) = sin(Z)/cos(Z)$ $arctg(Z) = (1/(2\cdot i)) \cdot ln((1+i\cdot Z)/(1-i\cdot Z))$

$ctg(Z) = cos(Z)/sin(Z)$ $arcctg(Z) = \pi /2-(1/(2\cdot i)) \cdot ln((1+i\cdot Z)/(1-i\cdot Z))$

$th(Z) = sh(Z)/ch(Z)$ $arcth(Z) = 0.5\cdot ln((1+Z)/(1-Z))$

$cth(Z) = ch(Z)/sh(Z)$ $arccth(Z) = 0.5\cdot ln((Z+1)/(Z-1))$

$sh(Z) = 0.5\cdot(exp(Z)-exp(-Z))$ $arcsh(Z) = ln(Z \pm (Z^2 +1)^{0.5})$

$ch(Z) = 0.5\cdot(exp(Z)+exp(-Z))$ $arcch(Z) = ln(Z \pm (Z^2-1)^{0.5})$

$arcsin(Z) = \pi /2-i\cdot ln(Z+(Z^2-1)^{0.5})$ $arccos(Z) = i\cdot ln(Z \pm (Z^2-1)^{0.5})$

Для функций, при вычислении которых используется CLOG(Z), определяются их главные значения.

В арифметическом выражении совместно с комплексными операндами могут быть использованы величины любого типа. При вычислении выражений со смешением типов применяются правила преобразования типов. Так, при выполнении арифметической операции между операндами, один из которых является комплексным, результат будет также комплексным. Запрещены арифметические операции между комплексными величинами и вещественными удвоенной точности. При выполнении оператора присваивания тип вычисленного значения в правой части оператора преобразуется к типу переменной в левой части.

При вводе-выводе комплексных данных необходимо задавать формат как для действительной части, так и для мнимой.

ЗАДАНИЕ 1. ВЫЧИСЛЕНИЕ КОМПЛЕКСНЫХ ФУНКЦИЙ. Для восьми значений комплексной переменной Z вычислить комплексные функции:

1) Z, |Z|, agr(Z), Re(Z), Im(Z), sin(Z), cos(Z), exp(Z), sqrt(Z), ln(Z) - для всех вариантов;

2) f1, f2, f3, f4 в соответствии с номером варианта из таблицы:

n	f1	f2	f3	f4
1	$Z^{2 \cdot i}$	$tg(Z) / th(1+i)$	$arctg((2- 0.5\cdot i)\cdot Z)$	$cos(-2\cdot i) \cdot ch(Z)$
2	$Z^{1/3}$	$ctg(i) / cth(Z)$	$arcctg((0.1+ i)\cdot Z)$	$sin(0.1\cdot i) \cdot sh(Z)$
3	$(2+i)^{Z}$	$th(Z) / tg(3- i)$	$arcsin((-1- i)\cdot Z)$	$(2-i) \cdot sh(i\cdot Z)$
4	$(i\cdot Z)^{1/4}$	$(3-0.4\cdot i)\cdot th(Z)$	$arccos(0.1-0.3\cdot i\cdot Z)$	$ch((0.2-i)-2\cdot Z)$
5	$(1+Z)^{2 \cdot i}$	$cth(0.5\cdot i \cdot (Z+i))$	$arcth(Z / (2- i))$	$sh((1- i)\cdot(Z+1))$
6	$(2\cdot i+Z)^{1/Z}$	$th(Z) / th(-i)$	$arccth(i / Z)$	$ch(Z+i) / sin(i)$
7	$Z^{i/2}$	$ctg(Z / (2+0.1\cdot i))$	$arcsh(i + i / Z)$	$sh(i -Z)\cdot cos(i)$
8	$(1+Z)^{1/2}$	$tg(Z / (1+Z))$	$arcch(Z / (1+ i))$	$sh(Z)\cdot ch(1+Z)$
9	$(i+Z)^{1/2}$	$th((2-i) \cdot Z)$	$arctg((2- i)\cdot Z)$	$sh(Z/(1-i))$
10	Z^{1+i}	$th(Z / (3+ i))$	$arcctg(Z\cdot(1-2\cdot i))$	$ch(Z\cdot(Z+1))$
11	$(2\cdot i+Z)^{0.5}$	$th(1+i) / tg(Z)$	$arcsin((Z+ i) / Z)$	$cos(i)\cdot ch(Z)$
12	$(1-i)^{Z+1}$	$cth(Z) / ctg(i)$	$arccos(Z / (Z - i))$	$sh(i)\cdot cos(Z)$
13	2^{1+i+Z}	$th(i-Z) / ctg(i)$	$arcth(Z^2 +1)$	$sh(i)\cdot sin(Z)$
14	$(Z+1)^{Z - i}$	$cth(2\cdot i) / tg(i\cdot Z)$	$arccth(Z+Z / i)$	$cos(Z+1)/ch(Z)$

15	$(-1+i)^{Z}$	$ctg(1-i)\cdot th(Z)$	$arcsh(Z/(Z+2))$	$\sin(2-i)\cdot ch(Z)$
16	Z^{-1-i}	$th(Z+1)/th(Z+i)$	$arcch((1-i)/Z)$	$\cos(2-i)\cdot ch(Z)$

Значения комплексной переменной Z_k вычислить в соответствии с номером n варианта:
$Z_1 = n$; $Z_2 = 1+ i\cdot n$; $Z_3 = i\cdot n$; $Z\,4= - n+i$; $Z_5 = - n$; $Z_6 = - n/3-i\cdot 2\cdot n/6$; $Z_7 = -i\cdot n$; $Z_8 = 2\cdot n/3-i\cdot 4\cdot n/5$

ЗАДАНИЕ 2. КОНФОРМНОЕ ОТОБРАЖЕНИЕ. Для области D в плоскости Z построить отображенную функцией W= f(Z) область D* в плоскости W. Для этого:

1. записать уравнения, описывающие границы области D;
2. выбрать (вычислить) на границах области D не менее 20 точек Zк таким образом, чтобы при их обходе область оставалась всегда слева;
3. вычислить точки Wк = f(Zк) области D*;
4. по найденным точкам на миллиметровке построить области D и D*.

Варианты заданий.

N	D	W
1	$\lvert Z\rvert < 1$	$(Z+2)/(Z+i)$
2	$-\pi/2 < Re(Z) < \pi/2$	$tg(Z)$
3	$\lvert Z\rvert > 3$	$(Z+1/Z)/2$
4	$-\pi/4 < Im(Z) < \pi/4$	$th(Z)$
5	$\lvert Z\rvert < 0.5$	$(Z+1/Z)/2$
6	$0 < \lvert Z\rvert < 1,\ 0 < \arg(Z) < \pi/3$	$(Z+1/Z)$
7	$\lvert Z-1\rvert < 1$	$1/Z$
8	$\lvert Z\rvert > 1,\ 0 < \arg(Z) < \pi/2$	$i\cdot \ln(Z)$
9	$\lvert Z-2\rvert > 2$	$1/(Z-1)$
10	$0 < Im(Z) < 2\pi,\ Re(Z) < 0$	$i\cdot \exp(Z)$
11	$0 < Im(Z) < \pi$	$ch(Z)$
12	$0 < Re(Z) < \pi/2,\ 0 < Im(Z) < 1$	$\exp(2i\cdot Z)$
13	$0 < Re(Z) < \pi/4$	$ctg(Z)$
14	$-\pi < Re(Z) < \pi,\ Im(Z) > 0$	$\sin(Z)$
15	$-\pi < Im(Z) < \pi$	$Z+\exp(Z)$
16	$Re(Z) > 0,\ -1 < Im(Z) < 0$	$ch(Z)$

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра прочности летательных аппаратов

Паспорт
расчетно-графического задания (работы)

по дисциплине «Практикум по алгоритмизации и программированию», 4 семестр

1. **Методика оценки**

В рамках расчетно-графического задания (работы) по дисциплине студенты должны рассчитать на изгиб шарнирно-опертую балку.

При выполнении расчетно-графического задания (работы) студенты должны составить алгоритмы и программы вычисления перемещений, углов поворота, моментов и перерезывающих сил балки. Студенты составляют программы, выполняют расчеты и строят эпюры с использованием программных продуктов **Microsoft Excel, MathCAD, Fortran**. Отчет оформляется в Word.

Обязательные структурные части РГЗ.

Изображение расчетной схемы в масштабе.
Разработка алгоритма построения сетки для разрывной функции.
Табулирование функций по участкам.
Определение констант из граничных условий.
Построение эпюр по вычисленным значениям функций.
Качественное оформление РГР.

2. **Критерии оценки**

- Работа считается **не выполненной,** если выполнены не все части РГЗ(Р), отсутствует описание алгоритмов и программ, допущены грубые ошибки в результатах, оценка составляет менее 0,25 максимального балла, указанного в БРС.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ(Р) выполнены формально: не приведено описание алгоритмов, допущены неточности в результатах, оценка составляет более 0,25 максимального балла, указанного в БРС.
- Работа считается выполненной **на базовом** уровне, если содержит все обязательные структурные части, не содержит неточностей, оценка составляет менее 0,75 максимального балла, указанного в БРС.
- Работа считается выполненной **на продвинутом** уровне при качественном оформлении и безошибочном выполнении всех структурных частей РГР, оценка составляет более 0.75 балла, указанного в БРС.

3. **Шкала оценки**

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. **Примерный перечень тем РГЗ(Р)**

Пример выполнения варианта задания

1. РАСЧЕТ НА ИЗГИБ ШАРНИРНО-ОПЕРТОЙ БАЛКИ

1.1 Постановка задачи

Задана балка, закрепленная на двух шарнирных опорах, и нагружена в соответствии с расчетной схемой (рис. 1.1). Для данной балки построить эпюры прогибов, углов поворота, моментов и перерезывающих сил, используя три программы Fortran, Mathcad, Excel. Исходные данные приведены в таблице 1.1.

Исходные данные. Таблица 1.1.

Параметры	*Значение параметров*
M_1, кНм	*40*
P_1, кН	*-10*
P_2, кН	*-20*
P_3, кН	*10*
q, кН/м	*-10*
a, м	*0,5*
EI	*1*

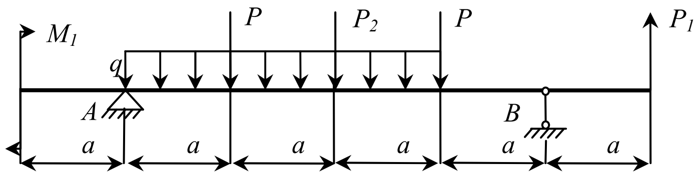


Рис. 1.1 Шарнирно-опертая балка

1.2 Разработка алгоритма вычисления функции прогиба

Изменим расчетную схему на рис. 1.1. На новой схеме (рис. 1.2) направим все силы и все реакции вверх. Распределенную нагрузку продолжим до конца, т.к. она должна заканчиваться на правом конце балки, и уравновесим ее другой распределенной нагрузкой, направленной в противоположную сторону.

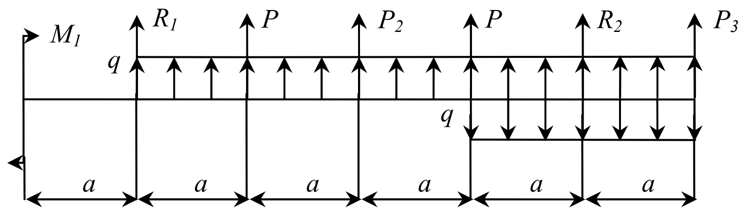


Рис. 1.2 Расчетная схема

Выберем систему координат так, чтобы начало ее совпадало с левым концом балки (рис. 1.2), разобьем балку на силовые участки.

Текст программы в Mathcad:

Для начала введем все исходные данные из таблицы 1.1. В Mathcad’е существует операция присваивания, которая позволяет оценивать любую переменную справа от знака присваивания и назначает любой результат слева от этого знака.

$M1 := 40\quad P1 := -10\quad EI := 1\quad P2 := -20\quad q := -10\quad a := 0.5\quad P3 := 10$

Из условия равновесия найдем реакции опор в точках *A* и *B*:

$$R1 := \frac{-M1 - 5q\cdot a\cdot 1.5\cdot a - P1\cdot 3\cdot a - P2\cdot 2\cdot a - P1\cdot a + P3\cdot a}{4a},\quad R1 = 11.875,$$

$$R2 := \frac{M1 - 5q \cdot a \cdot 2.5 \cdot a + 2q \cdot a \cdot 4 \cdot a - P1 \cdot a - P2 \cdot 2a - P1 \cdot 3 \cdot a - P3 \cdot 5a}{4a}, \quad R2 = 33.125.$$

Далее вводим функцию z, которая должна удовлетворять условию: $z(x,a) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases}$. Это необходимое условие, для того чтобы универсальное уравнение было верно для каждого участка.

В Mathcad это будет иметь вид:

$z(x,a) := \text{if}(x < a, 0, 1).$

Затем вводим функцию *W(x)*, которая будет входить в универсальное уравнение упругой линии, для определения прогибов и углов поворота. Эта функция должна быть больше либо равна нулю, т.к. здесь учитываются силовые факторы, лежащие слева от рассматриваемого сечения.

$$W(x) := M1 \cdot \frac{x^2}{2} + R1 \cdot \frac{(x-a)^3}{6} \cdot z(x,a) + P1 \cdot \frac{(x - 2 \cdot a)^3}{6} \cdot z(x, 2 \cdot a) + P2 \cdot \frac{(x - 3 \cdot a)^3}{6} \cdot z(x, 3 \cdot a) \ldots$$
$$+ P1 \cdot \frac{(x - 4 \cdot a)^3}{6} \cdot z(x, 4 \cdot a) + R2 \cdot \frac{(x - 5 \cdot a)^3}{6} \cdot z(x, 5 \cdot a) + P3 \cdot \frac{(x - 6 \cdot a)^3}{6} \cdot z(x, 6 \cdot a) \ldots$$
$$+ q \cdot \frac{(x-a)^4}{24} \cdot z(x,a) - q \cdot \frac{(x - 4 \cdot a)^4}{24} \cdot z(x, 4 \cdot a).$$

Введем матрицы D и B:

$$D := \begin{pmatrix} 1 & a \\ 1 & 5a \end{pmatrix}, \quad B := \begin{pmatrix} \frac{-1}{EI} \cdot W(a) \\ \frac{-1}{EI} \cdot W(5a) \end{pmatrix}, \quad B = \begin{pmatrix} -5 \\ -125.026 \end{pmatrix},$$

где **D** – матрица коэффициентов при неизвестных, а **B** – матрица свободных коэффициентов.

Эти матрицы нужны для того, чтобы решить СЛАУ **DC=B**, где **C** – неизвестные коэффициенты.

Для решения СЛАУ используем оператор lsolve

$$C := \text{lsolve}(D, B), \quad C = \begin{pmatrix} 25.007 \\ -60.013 \end{pmatrix}.$$

Задаем функцию, определяющую прогиб:

$$V(x) := C_0 + C_1 \cdot x + \frac{1}{EI} \cdot W(x).$$

Далее строим эпюру прогибов *V(x)*, выбрав на панели инструментов график в декартовых осях. Слева от оси ординат записываем *V(x)*, а на оси абсцисс устанавливаем пределы для координаты x от 0 до 3.

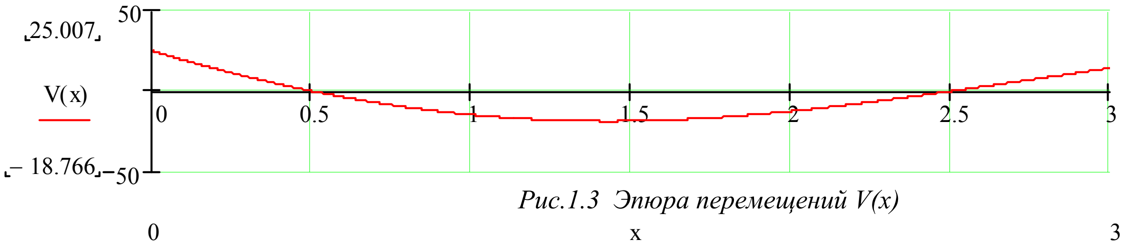


Рис.1.3 Эпюра перемещений V(x)

Продифференцируем в ручную *V(x)* и получим функцию для углов поворота *Q(x)*:

$$Q(x) := C_1 + M1 \cdot x + R1 \cdot \frac{(x-a)^2}{2} \cdot z(x,a) + P1 \cdot \frac{(x-2 \cdot a)^2}{2} \cdot z(x, 2 \cdot a) + P2 \cdot \frac{(x-3 \cdot a)^2}{2} \cdot z(x, 3 \cdot a) \ldots$$
$$+ P1 \cdot \frac{(x-4 \cdot a)^2}{2} \cdot z(x, 4 \cdot a) + R2 \cdot \frac{(x-5 \cdot a)^2}{2} \cdot z(x, 5 \cdot a) + P3 \cdot \frac{(x-6 \cdot a)^2}{2} \cdot z(x, 6 \cdot a) \ldots$$
$$+ q \cdot \frac{(x-a)^3}{6} \cdot z(x,a) - q \cdot \frac{(x-4 \cdot a)^3}{6} \cdot z(x, 4 \cdot a)$$

Точно также как и для прогибов строим эпюру *Q(x)*:

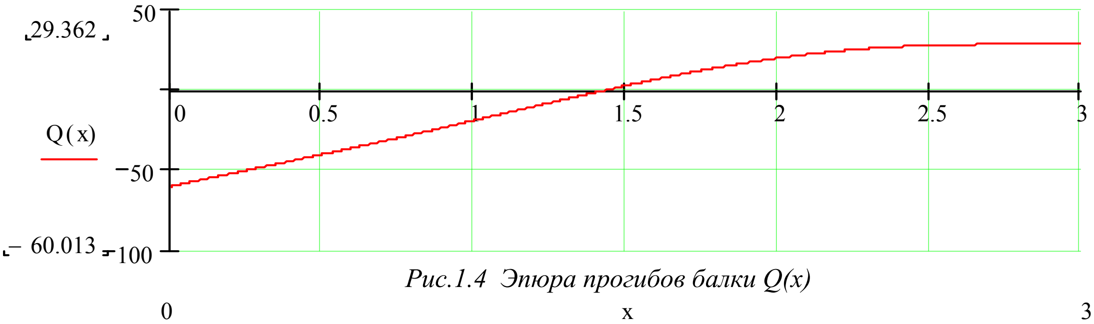


Рис.1.4 Эпюра прогибов балки Q(x)

Далее дифференцируя *Q(x)*, получаем функцию *M(x)*:

$$m(x) := M1 + R1 \cdot (x-a) \cdot z(x,a) + P1 \cdot (x - 2 \cdot a) \cdot z(x, 2 \cdot a) + P2 \cdot (x - 3 \cdot a) \cdot z(x, 3 \cdot a) \ldots$$
$$+ P1 \cdot (x - 4 \cdot a) \cdot z(x, 4 \cdot a) + R2 \cdot (x - 5 \cdot a) \cdot z(x, 5 \cdot a) + P3 \cdot (x - 6 \cdot a) \cdot z(x, 6 \cdot a) \ldots$$
$$+ q \cdot \frac{(x-a)^2}{2} \cdot z(x,a) - q \cdot \frac{(x-4 \cdot a)^2}{2} \cdot z(x, 4 \cdot a)$$

Строим эпюру изгибающих моментов *M(x)*:

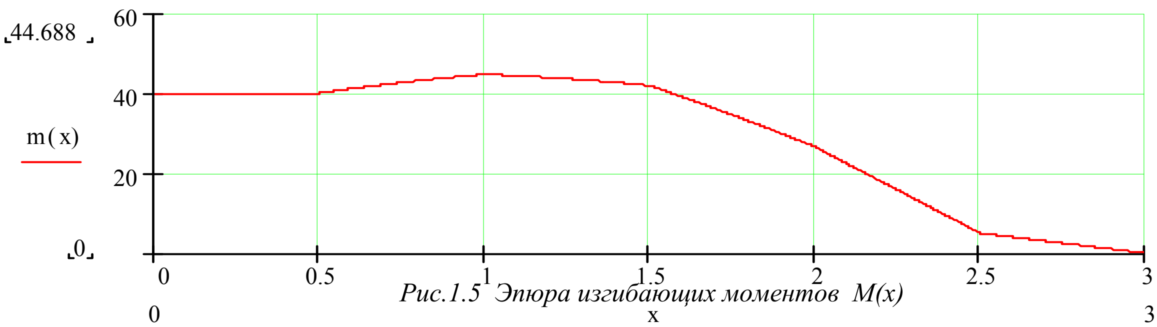


Рис.1.5 Эпюра изгибающих моментов M(x)

Дифференцируя *M(x)*, получаем *P(x)*:

$$P(x) := R1 \cdot z(x,a) + P1 \cdot z(x, 2 \cdot a) + P2 \cdot z(x, 3 \cdot a) \ldots$$
$$+ P1 \cdot z(x, 4 \cdot a) + R2 \cdot z(x, 5 \cdot a) + P3 \cdot z(x, 6 \cdot a) \ldots$$
$$+ q \cdot (x-a) \cdot z(x,a) - q \cdot (x - 4 \cdot a) \cdot z(x, 4 \cdot a)$$

Эпюра для перерезывающих сил *P(x)* имеет вид:

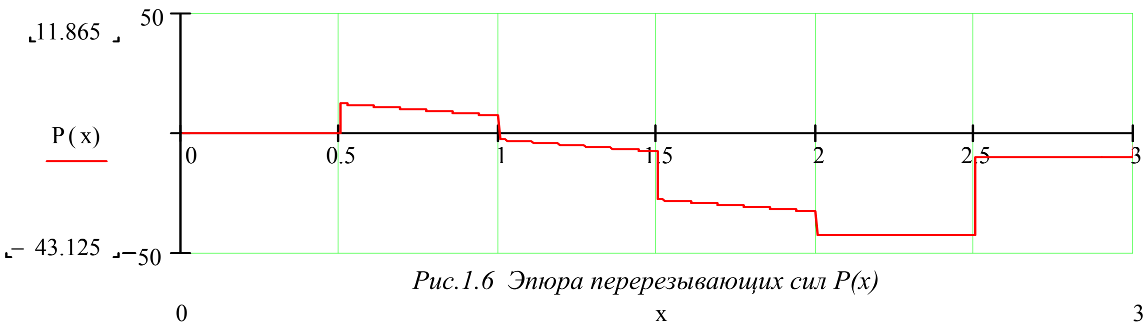


Рис.1.6 Эпюра перерезывающих сил P(x)

Как видно, текст программы очень прост и не требует особых комментариев. Меняя исходные данные, можно пользоваться этой программой повторно. Для построения эпюр других балок придется также изменять уравнения реакции опор.

Алгоритм решения задачи с использованием языка Fortran

1. Описываем заданные массивы
2. Описываем именованные константы, количество силовых участков и количество точек на участке.
3. Задаем функцию *z(i,j)* для каждого участка.
4. Задаем функцию *W(i)*, с использованием исходных данных.
5. Описываем массивы *V(nm)*, *W2(nm)*, *S(nm)*, *F(nm)*, *C*, *D*.
6. Вычисляем координаты узловых точек.
7. Вызываем подпрограмму SIMQ для решения СЛАУ
8. Вычисляем массивы
9. Выводим на экран искомые величины

Вызываем графическую подпрограмму для построения эпюр. Графическая подпрограмма позволяет визуально оценить полученный результат.

Программа на Fortran'е достаточно объемна, она состоит из головной программы и подпрограммы для построения самих эпюр, так же требуется вызывать подпрограмму SIMQ, для решения СЛАУ.

Эпюры, построенные на Fortran'е (рис. 1.7) схожи с эпюрами, построенными в Mathcad'е.

Различия между программами Mathcad и Fortran:

1. Fortran'е не существует различия между заглавными и прописными буквами, а в Mathcad'е идет четкое различие ввода переменных.

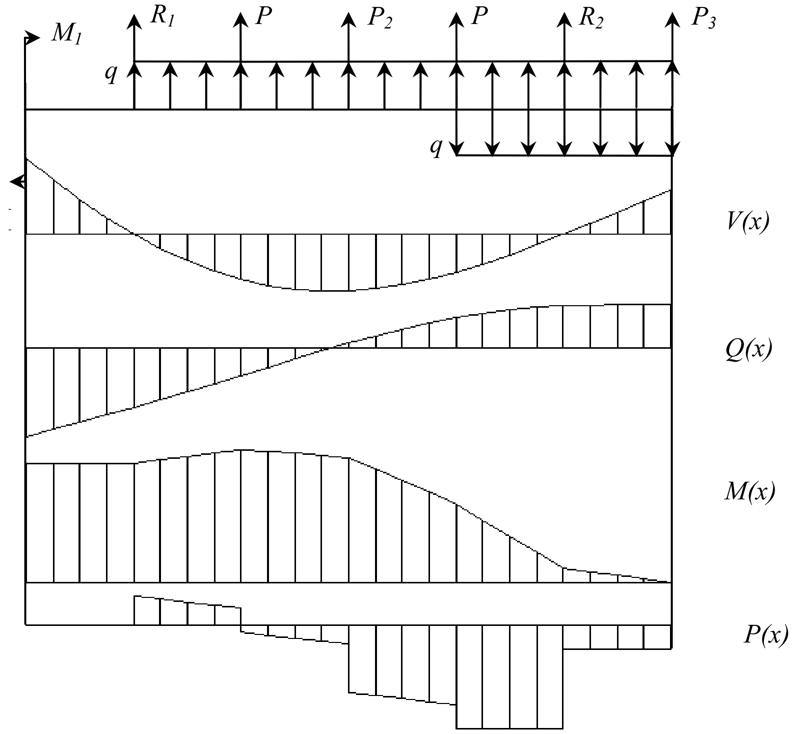
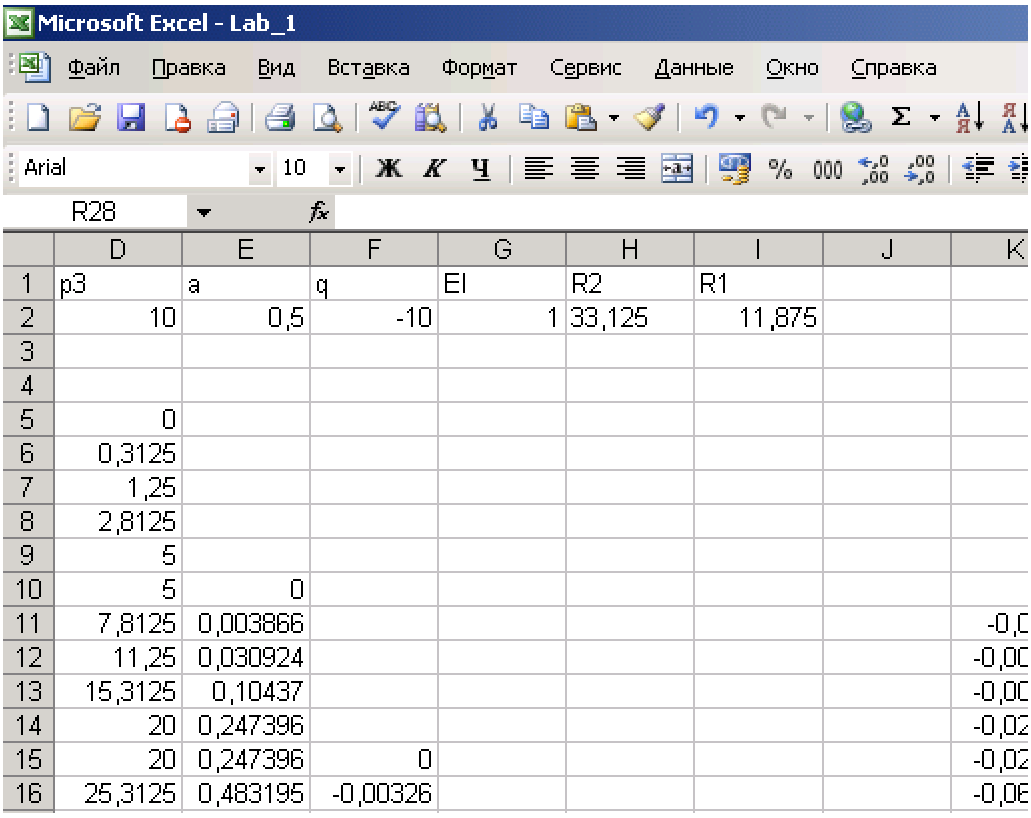


Рис.1.7 Эпюры построенные на Fortran'е

2. Mathcad не строит эпюры, а строит графики в отличии от Fortran.
3. Для построения эпюр на Fortran'е мы должны составлять подпрограмму, в отличие от Mathcad'а, где нужно только назначить операцию построения графика.
4. Текст программы в Mathcad'е значительно легче и проще для усвоения.

Построение эпюр в Excel

В начале вводим исходные данные для внешней нагрузки. Затем задаем шаг сетки по x и номеруем точки, всего точек будет 30. Далее в другом столбце задаем значение для x, в первой строке пишем 0, во второй предыдущее значению плюс шаг, при этом повторяем значения для узловых точек. Задаем функции для внешних нагрузок и затем суммируем их. Далее задаем матрицы **D** и **B**, обращаем матрицу **D** и умножаем ее на **B** для получения неизвестных коэффициентов. Задаем уравнение упругой линии, в которое входят полученные коэффициенты. Далее выделяя столбец со значениями x и столбец уравнения упругой линии, строим эпюру перемещений (рис. 1.8).



	D	E	F	G	H	I	J	K
1	p3	a	q	EI	R2	R1		
2	10	0,5	-10	1	33,125	11,875		
3								
4								
5	0							
6	0,3125							
7	1,25							
8	2,8125							
9	5							
10	5	0						
11	7,8125	0,003866						-0,
12	11,25	0,030924						-0,0
13	15,3125	0,10437						-0,0
14	20	0,247396						-0,0
15	20	0,247396	0					-0,0
16	25,3125	0,483195	-0,00326					-0,0

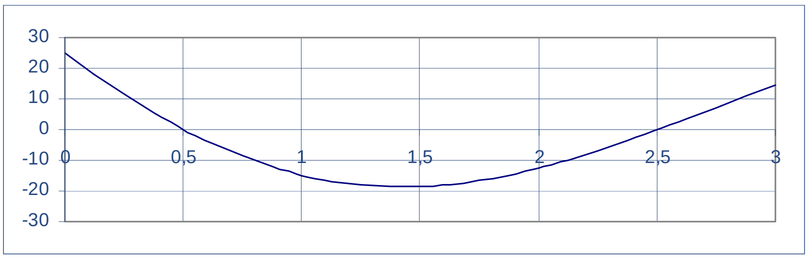


Рис.1.8 Эпюра прогибов балки Q(x)

Введение.

Стандартом Фортрана не предусмотрены средства графического вывода, построения диалогов.

Воспользовавшись приведенными в данной работе материалами, программист сможет организовать дружественный интерфейс (в виде диалоговых окон) между пользователем и выполняющими вычисления процедурами. Такие окна содержат управляющие элементы, поля ввода и вывода данных. Диалоги Фортрана обеспечивают полноценный интерфейс для большинства приложений. Но все же в случае сложной структуры интерфейса следует обратиться к другим программа (типа Delphi), предназначенных специально для разработки интерфейсов, так как создание такого интерфейса не целесообразно делать на Фортране.

Одновременно с диалогами в проектах QuickWin поддерживается многооконный графический вывод, управляемый из программы, из диалогов, из сопровождающего графические окна меню или процедурами, запускаемыми при нажатии на заданные клавиши клавиатуры или кнопки мыши. Имеющиеся в Фортране графические процедуры позволяют отображать растровые графические данные (в том числе и графический текст).

1. Использование диалогов.

1.1. Постановка задачи.

Исполняемый файл создадим как проект QuickWin. Такой проект создается в среде MS Developer Studio Fortran PowerStation 4.0 в результате выполнения цепочки:
File — New — Project Workspace — QuickWin Application — задать имя проекта (Name) и папку его размещения (Location) — Create.

Если осуществлять ввод данных в программе с экрана (использовать операторы print*, и read*,), то это может привести к тому, что, во-первых, такой ввод данных не нагляден, во-вторых, русский текст в DOS-окне претерпит искажения и, в-третьих, отображаемые в окне данные после перехода на следующую строку окна не могут быть отредактированы. Все эти недостатки можно устранить, создав диалог ввода данных.

1.2. Построение диалогового окна.

Диалог, после того как ясен его состав и геометрия расположения полей с данными и текстовых полей, создается в три этапа:
1) построение диалогового окна;
2) включение диалога в проект;
3) создание процедур инициализации и обработки полей диалога.

1.2.1. Проект для диалога.

Реализуем диалоговое окно, приведенное на рис.1.1.

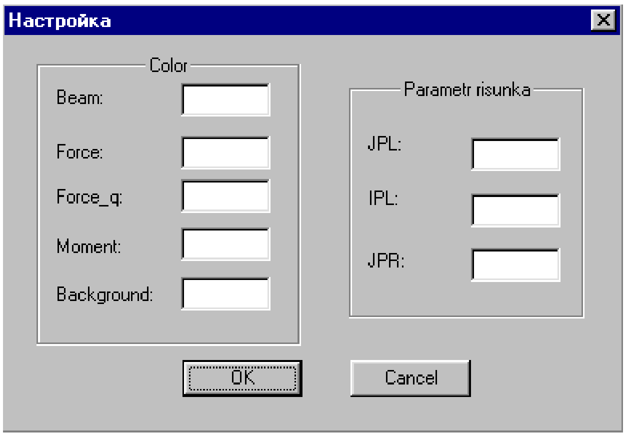


Рис.1.1. Проект диалогового окна.

Построение такого окна выполним в среде MS Developer Studio. Загрузим заготовку для диалога: Insert — Resource — выбрать Dialog — OK. В результате появится заготовка для окна с двумя кнопками — OK и Cancel.

1.2.2. Задание параметров диалога.

Настроим прежде параметры всего окна, щелкнув для этого 2 раза мышью по свободной поверхности окна. В появившемся окне Dialog Properties на вкладке General зададим:

1) в поле ID имя диалога — IDD_tun, по которому он будет идентифицироваться в программе;
2) в поле Caption название диалога — “Настройка”, которое отображается на верхней полосе диалога;
3) установим, нажав на кнопку Front, шрифт MS Sans Serif, и его размер, например 8;
4) в полях XPos и YPos координаты верхнего левого угла диалога, которые он имеет в дочернем окне QuickWin, например 150 и 100 (координаты задаются в текстовых столбцах и рядах).

Выберем затем вкладку Styles и оставим стоять галочки в пунктах Titlebar и System menu.

Остальные поля этой и других вкладок оставим без изменений (рис.1.2.).

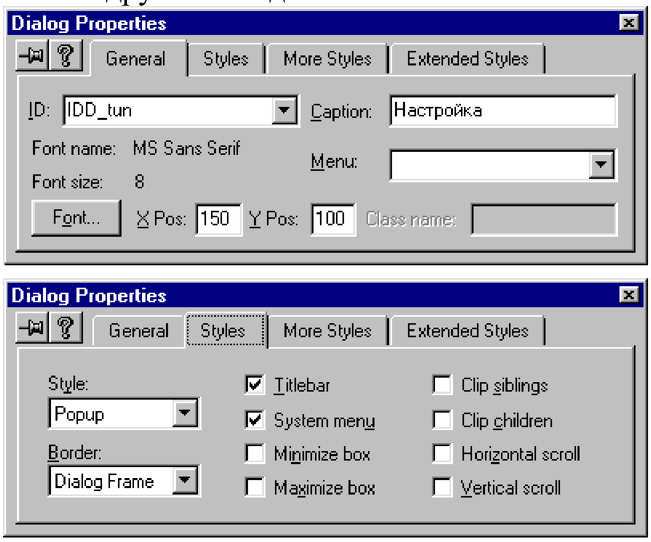


Рис.1.2. Параметры диалогового окна.

Закроем окно Dialog Properties, нажав, например, на клавишу Enter; увеличим привычным образом размеры диалогового окна; нажмем Ctrl+T и посмотрим получившееся окно. Нажмем Esc, с тем чтобы затем продолжить построение.

1.2.3. Задание и обработка статического текста.

Используя меню с управляющими элементами (рис.1.3.), зададим текстовые поля, группы и поля ввода и редактирования данных.

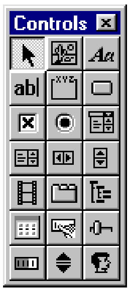


Рис.1.3. Управляющие элементы.

Задание текстового поля (Static text) выполняется после выбора управляющего элемента ***Aa*** и последующей прорисовки прямоугольника для предполагаемого текста в свободной области диалога. Прорисовка выполняется мышью при нажатии ее левой клавише. (Можно также после выбора элемента управления просто “ударить” мышью по предполагаемому его месту размещения в диалоге.) Выделим получившейся прямоугольник, ударив по нему мышью, и изменим его положение и размеры, пользуясь мышью или клавиатурой. Так, увеличит размер по оси *x* можно, нажимая Shift+→. “Ударим” теперь по тексту 2 раза мышью и зададим в окне Text Properties во вкладке General свойство Caption текста: “Beam:” (рис.1.1.). Остальные свойства оставим без изменений.

Аналогичным образом зададим все иные тексты, приведенные на рис 1.1. Все тексты, поскольку они неизменяемы, будут иметь один и тот же задаваемый по умолчанию идентификатор (свойство ID): IDC_STATIC (или можно задать свой идентификатор: IDC_Col1).

Напомним, что для отмены действия можно использовать Ctrl+Z.

Задаваемый подобным образом текст не может быть изменен в диалоге.

1.2.4. Обработка редактируемых полей.

Выберем контрольный элемент ***ab|*** и, применяя ту же технику, что и при создании статического текста, разместим в диалоге 8 редактируемых полей (Edit Box). Используем первое поле для ввода цвета балки, второе — для ввода цвета стрелок сил и т.д. (см. рис.1.1.) Присвоим полям имена: первому — *Col1_ed*, второму — *Col2_ed* и т.д. Имя поля задается в связанном с ним окне Edit Properties изменения свойств поля, которое появляется, если по полю дважды щелкнуть мышью. По умолчанию ID-имя первого поля имеет значение IDC_EDIT1, которое мы изменим на *Col1_ed*. Аналогичные операции повторим и для остальных редактируемых полей.

Перемещение по редактируемым полям и другим элементам управления диалога осуществляется при помощи мыши, клавиш Tab, Shift+Tab, Enter (Ввод и перемещение) и других клавиш.

Редактируемые поля содержат символьные данные. Для передачи из программы в символьные поля числовых данных выполняются преобразования “число — строка”. При вводе числовых данных из символьного редактируемого поля диалога выполняются преобразования “строка — число”.

Изменение значения поля при открытом диалоге можно выполнить из программы, применив функцию DLGSET. Функция DLGSET имеет логический тип и вернет .TRUE. в случае успеха. Например, после вызова

```
l4=dlgset(dlg,idc_col1_ed,c4)
```

в поле *Col1_ed* появится значение с4.

Передача данных из поля в программу выполняется в результате вызова функции DLGGET. Например, вызова

```
l4=dlgget(dlg,idc_col1_ed,c4)
```

обеспечит передачу значения поля *Col1_ed* в символьную переменную *string*. Функция DLGGET имеет логический тип и вернет .TRUE. в случае успеха.

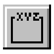
Напомним, что последующее преобразование “строка — число” выполняется так:

```
read(c4,fmt4)cvb
```

Преобразование “число — строка” моно выполнить так:

```
write(c4,fmt4)cvb.
```

1.2.5. Задание и обработка группы.

Задание группы (Group Box) выполняется после выбора управляющего элемента  и последующей прорисовки прямоугольника для предполагаемой группы, объединяя в группу элементы статического текста (“Beam”, “Force”, “Force_q”, “Moment”, “Background”) и редактируемых полей, соответствующих своим текстам. Прорисовка выполняется мышью при нажатии ее левой клавише. Выделим получившейся прямоугольник, ударив по нему мышью, и изменим его положение и размеры, пользуясь мышью или клавиатурой. “Ударим” теперь по тексту 2 раза мышью и зададим в окне Group Box Properties во вкладке General свойство Caption текста: “Color” (рис.1.1.). Выберем затем вкладку Styles и в пункте Horizontal alignment: поставим Center. Остальные свойства оставим без изменений.

Аналогичным образом зададим остальную группу. Все группы, поскольку они неизменяемы, будут иметь один и тот же задаваемый по умолчанию идентификатор (свойство ID): IDC_STATIC (или можно задать свой идентификатор: IDC_Col).

Задаваемая подобным образом группа не может быть изменена в диалоге.

1.2.6. Кнопки OK и Cancel.

Кнопки OK и Cancel появляются в каждом вновь создаваемом диалоге, и с ними связаны задаваемые по умолчанию процедуры. Можно изменить как ID-имена этих кнопок, так и названия. Разумеется, каждая из этих кнопок может быть удалена из диалога.

1.2.7. Меню диалога.

Помимо приведенного на рис.1.3. меню с управляющими элементами (Controls) при работе с диалогом можно пользоваться меню Dialog (рис.1.4.).

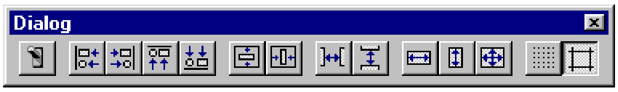


Рис.1.4.Меню диалога.

Это меню позволяет изменять положение и геометрию выбранных полей диалога, задавать сетку в проектируемом окне, линейку измерения размеров, просматривать (Ctrl+T) диалог.

Если выбрано одно поле, то его можно отцентрировать как по вертикали (Ctrl+F9), так и по горизонтали (Ctrl+Shift+F9). Если выбраны несколько полей, то они могут быть:

- выровнены влево (Align Left-Ctrl+←);
- выровнены вправо (Align Left-Ctrl+→);
- выровнены вверх (Align Top-Ctrl+↑);
- выровнены вниз (Align Bottom-Ctrl+↓);
- отцентрированы по вертикали и горизонтали;
- сделаны одной ширины (Make Same Width);
- сделаны одной высоты (Make Same Height);
- сделаны одного размера (Make Same Size).

Меню диалога можно убрать, нажав, например, на крест (×) (рис.1.4.). При необходимости можно восстановить, найдя в главном меню пункт Toolbars и поставив галочку после выбора Toolbars рядом с пунктом Dialog.

Аналогичным образом можно активизировать и ранее закрытое меню Controls.

1.2.8. Меню диалога.

При сохранении диалога создается файл ресурсов, которому мы присвоим имя iin.rc и который расположим в папке, содержащей исходный код. При сохранении диалога в папке с файлом iin.rc появляются или обновляются файлы resource.h и resource.fd, содержащие параметры диалога и других ресурсов.

Файл ресурсов может содержать диалоги, меню и другие ресурсы, однако пока в нем только один диалог — IDD_tun. Файл iin.rc, чтобы программа получила к нему доступ, необходимо вставить в проект. Выполним в FPS цепочку: Insert — File into Project — задать Resource files для типа файлов — выбрать файл iin.rc — Add.

Выполним компиляцию проекта. В среде MS Developer Studio — в окне работы с проектом (View window) — появится вкладка Resource view, переместившись в которую можно просмотреть и состав файла ресурсов iin.rc, и отдельно каждый из его компонентов. Существующее диалоговое окно будет загружено в среду MS Developer Studio, если дважды щелкнуть мышью по его ID-имени, отображаемому на вкладке Resource view. Появившееся окно доступно для редактирования.

На вкладке File View после компиляции проекта появится подраздел Dependencies, содержащий ссылку на файл resource.fd. Отметим, что пользователь должен воздержаться от ручного редактирования этого файла.

1.2.9. Работа с диалогом в программе.

программу реализуем по следующей схеме:

- выполним инициализацию диалога;
- отобразим, обратившись к функции DLGMODAL, диалог на экран;
- введем данные;
- нажмем кнопки OK и Cancel;
- при выборе OK или Cancel закроем диалог и завершим программу.

Функция DLGINIT имеет логический тип и вернет .TRUE. в случае успеха. Функция DLGINIT выполняет инициализацию диалога.

Функция DLGMODAL типа INTEGER(4) в случае успеха отображает диалог на экране и передает управление диалогу. В случае неудачи функция вернет –1. Функция имеет синтаксис:

result=DLGMODAL(*dlg*)

dlg — переменная типа *dialog*, связанная посредством функции DLGINIT с конкретным диалогом.

Подпрограмма DLGEXIT закрывает диалог и имеет синтаксис:

CALL DLGEXIT(*dlg*)

Закрытому по средством DLGEXIT диалогу можно передать управление, применив функцию DLGMODAL.

1.3. Управляющие элементы построенного диалога.

Таблица 1.1. Управляющие элементы для данного диалога (рис.1.1.)

Управляющий элемент	*Описание*
Статический текст	Изменяемый из программы текст
Редактируемое поле	Элемент, значение которого можно изменить с клавиатуры
Группа	Элемент, используемый для объединения элементов, в группу взаимодействующих элементов

1.4. Процедуры для работы с данным диалогом.

В FPS интерфейсы функций размещены в модуле DIALOGM.MOD.

Таблица 1.2. Процедуры для данного диалога (рис.1.1.)

Процедура	*Тип*	*Назначение*
DLGEXIT	Подпрограмм а	Закрывает открытый диалог
DLGGET	LOGICAL(4)	Передает из диалога в программу значение компонента управляющего элемента
DLGINIT	“	Инициализация диалога
DLGMODA L	INTEGER(4)	Отображает диалог на экране и передает управление диалогу
DLGSET	LOGICAL(4)	Устанавливает значение компонента управляющего элемента

2. Вывод графических данных.

Графические процедуры могут быть использованы в проектах QuickWin (многооконный графический проект).

2.1. Графический дисплей.

В современных ЭВМ используются растровые дисплеи. Графические возможности растрового дисплея определяются размером экрана, зерном экрана, частотой регенерации и разрешением.

Размер экрана может варьироваться от 14 до 21 дюйма и более.

Вывод изображения выполняется в результате подсветки электронным лучом отдельных регулярно расположенных точек экрана. Такая точка является наименьшим элементом графической информации и называется *пикселем*.

Число таких точек по горизонтали и вертикали экрана определяется его *разрешением*. Стандартными являются разрешения 640×480 (первая цифра указывает на число точек по горизонтали, вторая — по вертикали), 800×600, 1024×768, 1280×1024 и более высокие разрешения. Управление разрешением осуществляется графическим адаптером.

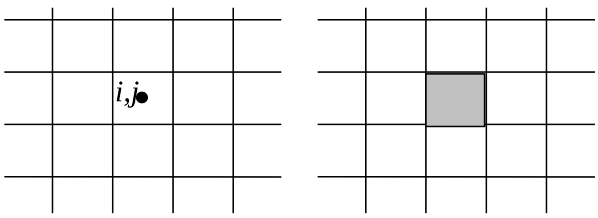
Луч последовательно "пробегает" все точки экрана, выполняя заданную программой для каждой точки подсветку. Затем цикл обхода повторяется. Полный цикл обхода всех точек экрана приводит к воспроизведению изображения и называется *регенерацией изображения*. Число регенерации в секунду называется *частотой регенерации*. Низкая частота регенерации приводит к заметному для человеческого глаза миганию изображения. Хорошим показателем можно считать частоту регенерации не менее 60 Гц.

Расстояние между соседними точками экрана определяет его *зерно*. Размеры зерна варьируются от 0,31 до 0,21 мм и менее. Понятно, что чем меньше зерно, тем выше качество изображения.

2.2. Растровое изображение.

В графике экран растрового дисплея представляется в виде прямоугольной сетки на дискретной плоскости с шагом по осям x и y, равным единице. Такая модель называется *растровой плоскостью* (рис.2.1.,а). Массив прямоугольных ячеек плоскости называется *растровым массивом*.

Каждый квадрат сетки соответствует одному пикселю экрана. Точка инициализации пикселя находится в центре квадрата сетки. Инициализация точки растровой плоскости с координатами (i,j) соответствует закраска квадрата сетки, в центре которого эта точка расположена (рис.2.1.,б).



а б

Рис.2.1. Модель растрового экрана:
а — растровая плоскость;
б — инициализация точки растровой плоскости.

Растровое изображение создается путем закраски ячеек растрового массива в тот или иной цвет. Растровое изображение можно сравнить с изображением на листе клетчатой бумаги, получаемым в результате закраски отдельных клеточек листа.

Каждому пикселю растровой плоскости могут быть независимым образом заданы цвет, интенсивность и другие характеристики.

2.3. Видеоадаптер.

Данные, подлежащие воспроизведению на экране монитора, формируются программой, функционирующей на центральном процессоре ЭВМ. Далее они передаются видеоадаптеру, который размещает эти данные в видеопамяти, преобразовывает и передает устройству управления лучом электронно-лучевой трубке.

Таким образом, *видеоадаптер* является устройством сопряжения центрального процессора ЭВМ и устройства отображения.

Видеоадаптеры могут обеспечить разные режимы отображения данных, которые разделяются на *текстовые* и *графические*. Причем, как правило, видеоадаптер поддерживает несколько графических режимов. Режим отображения данных на экране называется *видеорежимом*. В MS Developer Studio видеорежим устанавливается при задании конфигурации видеоокна функцией SETWINDOWCONFIG.

Различные видеорежимы отличаются количеством выводимых на экран данных и цветов. Единицами измерения количества выводимых данных являются: в текстовом режиме — символ (литера), в графическом — пиксель.

2.4. Видеоокно и окна вывода.

В FPS в проектах с графикой вывод и текстовых и графических данных выполняется в *видеоокно*, параметры которого устанавливаются функцией SETWINDOWCONFIG. В проектах со стандартной графикой получаемое изображение отображается на полном экране без меню и вертикальной и горизонтальной полос прокрутки экрана. Затем видеоокно получает необходимые для него меню и полосы прокрутки.

В QuickWin каждое создаваемое (дочернее) видеоокно всегда располагается внутри обрамляющего окна. Размеры и положение обрамляющего и любого дочернего видеоокна могут быть изменены не только пользователем, но и из программы.

Вывод текстовых данных в видеоокно выполняется операторами PRINT, WRITE и подпрограммой OUTTEXT, чтение — оператором READ.

Внутри окна могут быть созданы прямоугольные текстовые области, называемые *окнами вывода*. По умолчанию окном вывода является все видеоокно, на которое и направляется как графический, так и текстовый вывод. С этим окном связана физическая система координат. Единицей измерения значений текстовых координат является знакоместо, а графических — пиксель. Число создаваемых окон вывода произвольно.

Замечание. Далее, как правило, будем употреблять слово “окно” вместо более громоздкого “видеоокно”.

В каждом программном компоненте, где вызываются графические процедуры, необходимо выполнить ссылку на модуль MSFLIB (USE MSFLIB). Этот модуль содержит интерфейсы к графическим процедурам, необходимые для работы с графикой именованные константы и определения производных типов данных.

2.5. Задание конфигурации видеоокна.

установка конфигурации окна не является обязательной процедурой, но при необходимости может быть выполнена функцией

flag4=SETWINDOWCONFIG(*wc*)

wc — переменная производного типа *windowconfig*, содержащая параметры окна. Тип *windowconfig* определен в модуле MSFLIB.

Можно вообще не использовать SETWINDOWCONFIG. В этом случае окно выберет наилучшее из возможных разрешений и размер шрифта 8×16. Число доступных цветов зависит от видеоадаптера.

2.6. Системы графических координат. Окно вывода.

Растровая плоскость расположена в *физической системе координат*, определяемой техническими средствами. Начало физической системы координат — левый верхний угол окна. Ось абсцисс направлена слева направо; ось ординат — сверху вниз (рис.2.2.). Физические координаты целочислены (пиксели). Используемый при задании координат тип данных — INTEGER(2).

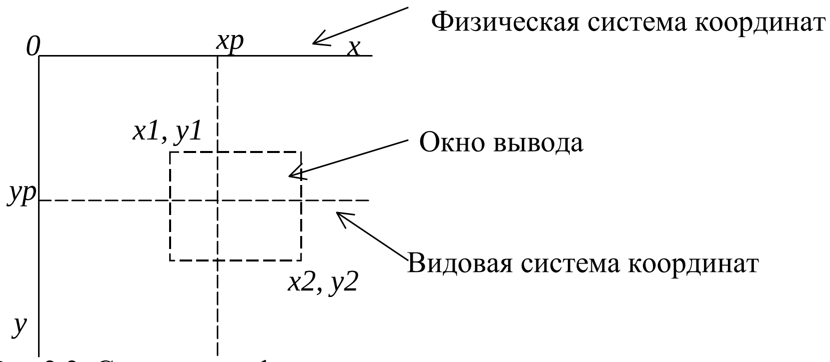


Рис.2.2. Системы графических координат.

Точка начала отсчета координат пикселей может быть перемещена в любую точку окна *xp, yp*. Эта процедура называется *заданием видового порта*. Так, после перемещения точки отсчета в центр видеоокна минимальные координаты пикселя при разрешении 800×600 будут равны (-400,-300), а максимальные — (399,299).

Видовой порт (видовую систему координат) и одновременно окно вывода можно задать одной подпрограммой

CALL SETVIEWPORT(*x1,y1,x2,y2*)

x1,y1 — координаты верхнего левого угла окна вывода;

x2,y2 — координаты нижнего правого угла окна вывода.

Тип параметров *x1,y1,x2,y2*— INTEGER(2).

Подпрограмма переопределяет графический видовой порт и задает окно вывода. То есть начало видовой системы координат после вызова

CALL SETVIEWPORT(*x1,y1,x2,y2*)

будет совпадать с верхним левым углом окна вывода.

Координаты (*x1,y1*) и (*x2,y2*) задаются в физической системе координат. Все последующие преобразования окна вывода посредством функции SETWINDOW выполняются в видовой, а не в физической системе координат.

Ранее рассмотрели системы с целочисленными координатами, диапазон изменения которых определяется разрешением дисплея и заданным окном. В FPS, однако, есть возможность задать окно вывода, связав его с *оконной системой координат* (ОСК), в которой используются вещественные координаты. Задаваемые в оконной системе координаты называются *оконными координатами*. ОСК может совпадать с *мировой системой координат*, т.е. с системой, в которой мы работаем с реальными объектами.

Окно вывода, в котором можно работать с вещественными координатами, задается в видовой системе координат функцией

result2=SETWINDOW(*finvert*,*wx1*,*wy1*,*wx2*,*wy2*)

finvert — параметр типа LOGICAL(2), определяющий направление координат (если *finvert* равен .TRUE., то ось *y* увеличивается от нижней границы видового порта к верхней; если *finvert* равен .FALSE., то ось *y* увеличивается от верхней границы видового порта к нижней, т.е. так же, как в видовой системе координат).

wx1,wy1 — верхний левый угол окна в ОСК;

wx2,wy2 — нижний правый угол окна в ОСК.

Тип параметров *wx1,wy1,wx2,wy2*— REAL(8). Тип функции — INTEGER(2). Функция возвращает отличное от нуля значение при успешном открытии окна и 0 в противном случае.

Функция SETWINDOW определяет ОСК, в которой координаты выводимых объектов ограничены значениями параметров *wx1,wy1,wx2,wy2*. В общем случае начало координат ОСК может находиться за пределами окна вывода.

В ОСК могут быть использованы для вывода графических элементов только функции, имена которых завершаются символами_W (например, LINETO_W).

Задание ОСК выполняется относительно текущего видового порта.

2.7. Очистка и заполнение экрана цветом фона.

Видеоокно, текущий видовой порт и текущее текстовое окно вывода очищаются после применения подпрограммы

CALL CLEARSCREEN (*area*)

area — определенная в модуле MSFLIB константа, указывающая на тип очищаемой области и принимающая значения:

$GCLEARSCREEN — очистка всего видеоокна;

$GVIEWPORT — очистка текущего видового порта;

$GWINDOW — очистка текущего текстового окна, заданного по средством SETTEXTWINDOW.

Помимо очистки заданная область заполняется цветом текущего фона.

2.8. Управление цветом.

2.8.1. Цветовая палитра VGA.

В режиме VGA с 16 цветами и разрешением 640× 480 пикселей по умолчанию устанавливается приведенная в табл.2.1. палитра цветов.

Таблица 2.1.Шестнадцатицветовая палитра VGA.

№	*Цвет*	*№*	*Цвет*
0	Черный	8	Серый
1	Синий	9	Светло-синий
2	Зеленый	10	Светло-зеленый
3	Голубой	11	Светло-голубой
4	Красный	12	Светло-красный
5	Фиолетовый	13	Светло-фиолетовый
6	Коричневый	14	Желтый
7	Белый	15	Ярко-белый

2.8.2. Управление цветом фона.

Задание текущего цвета фона по его номеру в цветовой палитре выполняется функцией

result4=SETBCOLOR(*index*)

index — параметр типа INTEGER(4), равный номеру цвета устанавливаемого фона.

Функция возвращает значение типа INTEGER(4), равное номеру цвета старого фона. По умолчанию устанавливается черный цвет фона.

Если же после вызова функции SETBCOLOR вызвать подпрограмму CLEARSCREEN, то выполниться очистка заданной области и ее заполнение заданным цветом.

2.8.3. Управление цветом графических примитивов.

Текущий цвет выводимых графических элементов: отрезка прямой, прямоугольника, многоугольника, эллипса, дуги, сектора — может быть задана функцией

result2=SETCOLOR(*index*)

index — параметр типа INTEGER(2), равный номеру устанавливаемого цвета.

2.9. Текущая позиция графического вывода.

Графические примитивы *текст* и *отрезок прямой* выводятся, начиная от текущей позиции. После перехода в графический режим текущая позиция находится в центре видеоокна. Изменение текущей позиции выполняется подпрограммой

CALL MOVETO_W (*wx,wy,wt*)

wx,wy — оконные координаты новой позиции типа REAL(8).

wt — оконные координаты предшествующей позиции (переменная типа *wxycoord*). Тип *wxycoord* определен в модуле MSFLIB.

При изменении текущей позиции изображение не меняется. Подпрограмма MOVETO_W изменяет текущую позицию в оконной системе координат.

2.10. Графические примитивы, используемые в данной работе.

Таблица 2.2.Процедуры вывода графических элементов.

Процедуры	*Отображаемые графические элементы*
LINETO_W	Отрезок прямой
RECTANGLE_W	Прямоугольник
POLYGON_W	Многоугольник
ELLIPSE_W	Эллипс или окружность
FLOODFIL_W	Заполнение (заливка) замкнутой области заданным цветом и по заданному шаблону

Для графического примитива могут быть установлены следующие характеристики:

- тип линии (например, сплошная, пунктирная);
- цвет линии;
- способ вывода(для отрезков, прямоугольников и многоугольников);
- шаблон и цвет заполнения (для замкнутых геометрических фигур).

Кроме того, вывод примитива может выполняться на заранее установленном фоне, т.е. после заливки окна вывода заданным цветом. По умолчанию цвет фона — черный, цвет линий — белый.

2.10.1. Вывод отрезка прямой линии.

Функция вывода отрезка прямой линии:

result2=LINETO_W(*wx*,*wy*)

wx,*wy* — оконные координаты конечной точки отрезка — REAL(8).

Функция возвращает значение типа INTEGER(2), которое, если отрезок нарисован, отлично от нуля или равно нулю в противном случае.

Отрезок выводится от текущей позиции до конечной точки, задаваемой параметрами *wx*,*wy*. При выводе отрезка используется текущий цвет, задаваемый функцией SETCOLOR. По умолчанию отрезки отображаются текущим цветом и выводятся сплошной линией

В случае успешного выполнения LINETO_W текущей становится соответствующая конечной точке позиция (*wx*,*wy*).

Замечание. Если область, границы которой образованы выведенными функцией LINETO_W отрезками прямых, заполняется посредством FLOODFIL_W, то для отображения границ области необходимо использовать сплошной тип линий.

2.10.2. Вывод прямоугольника.

Вывод прямоугольника выполняется функцией

result2=RECTANGLE_W(*control*, *wx1*,*wy1*,*wx2*,*wy2*)

control — флаг заполнения прямоугольника — INTEGER(2);

wx1,*wy1* — оконные координаты левого верхнего угла прямоугольника;

wx2,*wy2* — оконные координаты правого нижнего угла прямоугольника. Тип параметров *wx1*,*wy1*,*wx2*,*wy2*— REAL(8).

Функция возвращает значение типа INTEGER(2), которое, отлично от нуля при успешном выполнении или равно нулю в противном случае.

Границы прямоугольника рисуются с использованием текущего цвета, текущего способа вывода и текущего типа линии.

Параметр *control* может принимать значения определенных в модуле MSFLIB именованных констант $GFILLINTERIOR и $GBORDER.

Если *control* равен $GFILLINTERIOR, то осуществляется заполнение прямоугольника с использованием текущего цвета и шаблона заполнения. По умолчанию осуществляется сплошное заполнение замкнутой области.

Если *control* равен $GBORDER, то выводятся только границы прямоугольника с использованием текущих цветов, способа вывода и типа линии.

2.10.3. Вывод многоугольника.

Произвольной формы многоугольник выводиться функцией

result2= POLYGON_W (*control, wppoints, cpoints*)

Параметр *control* принимает те же значения и имеет тот же смысл и тип, что и одноименный параметр функции RECTANGLE_W.

wppoints — массив типа *wxycoord* из элементов, задающих ломанную линию;

cpoints — параметр типа INTEGER(2), задающий число точек ломанной линии.

Тип *wxycoord* определен в модуле MSFLIB.

Функция возвращает значение типа INTEGER(2), которое, отлично от нуля при успешном выполнении или равно нулю в противном случае.

Границы многоугольника вычерчиваются с использованием текущего цвета, текущего способа вывода и текущего типа линии.

Параметр *cpoints* указывает число используемых точек массива *wppoints*. Последняя выводимая точка массива всегда соединяется отрезком прямой с первой точкой.

2.10.4. Вывод эллипса и окружности.

Изображение эллипса и окружности выполняется функцией

result2= ELLIPSE_W (*control, wx1,wy1,wx2,wy2*)

control — флаг заполнения эллипса — INTEGER(2);

wx1,wy1 — оконные координаты левого верхнего угла ограничивающего эллипс прямоугольника;

wx2,wy2 — оконные координаты правого нижнего угла ограничивающего эллипс прямоугольника. Тип параметров *wx1,wy1,wx2,wy2*— REAL(8).

Функция возвращает значение типа INTEGER(2), которое, отлично от нуля при успешном выполнении или равно нулю в противном случае.

Границы эллипса вычерчиваются с использованием текущего цвета. Такие атрибуты, как способ вывода и тип линии, для эллипса не устанавливаются. Вывод эллипса всегда выполняется с применением сплошного типа линии.

Параметр *control* принимает те же значения и имеет тот же смысл и тип, что и одноименный параметр функции RECTANGLE_W и POLYGON_W.

2.11. Заполнение замкнутых областей.

Замкнутые области могут быть образованы одним графическим примитивом (прямоугольник, эллипс, многоугольник) или сочетанием нескольких, например отрезки прямых.

Как было сказано выше, графические примитивы: прямоугольник, эллипс, многоугольник — могут быть заполнены текущим цветом в процессе их построения, если управляющий заполнением параметр *control* равен $GFILLINTERIOR. При этом цвет заливки и границы примитива совпадают. Помимо этой возможности, любую замкнутую область, а не только отдельный примитив, можно заполнить текущим, установленным функцией SETCOLOR цветом, использовав функцию

result2= FLOODFIL_W (*wx,wy,bcolor*)

wx,wy — параметры типа REAL(8), задающие оконные координаты стартовой точки заполнения;

bcolor — параметр типа INTEGER(2), задающий номер цвета границы.

Функция возвращает отличное от нуля значение при успешном выполнении или 0 в противном случае. Последнее происходит, если заполнение не может быть завершено, например если стартовый пиксель (пиксель, в котором расположена стартовая точка) имеет цвет *bcolor* или если стартовая точка лежит за пределами окна вывода. Возвращаемое функцией FLOODFIL_W значение имеет тип INTEGER(2).

При заполнении области используется текущий цвет и текущая маска заполнения.

Если стартовая точка находится внутри области, то заполняется сама область. Если стартовая точка находится вне области, то заполняется пространство вне области. При задании стартовой точки на границе области заполнение не производится.

Заполнение осуществляется во всех направлениях, прекращаясь при достижении пикселей, окрашенных в цвет *bcolor* — цвет границы области.

Замечание. Заполнение области будет выполнено успешно, если область не имеет разрывов и для отображения ее границ использован сплошной тип линии.

3. Приложения QUICKWIN.

3. 1. Возможности QUICKWIN.

В Microsoft Fortran PowerStation (FPS) исполняемое приложение может быть создано как проект QuickWin — многооконный проект с графикой. В среде MS Developer Studio Fortran PowerStation 4.0 такой проект создается после выполнения цепочки: File — New — Project Workspace — QuickWin Application — задать имя проекта (Name) и папку его размещения (Location) — Create. В программных единицах, которые используют процедуры QuickWin, должна быть ссылка на модуль MSFLIB.MOD (в случае FPS):

```
USE MSFLIB
```

Работая с QuickWin, можно:

- скомпилировать консоль-приложение как простое приложение для Windows;
- минимизировать и максимизировать создаваемые в QuickWin окна, в которые выполняется передача данных;
- вызывать любую из приведенных в пункте 2 графическую процедуру;
- загружать и сохранять образы;
- выбирать, копировать и вставлять текст и графику;
- обнаруживать и обрабатывать удары мыши;
- выводить на принтер графические данные;
- программировать меню;
- создавать собственные иконы;
- открывать несколько дочерних окон.

Используя диалоговые окна и возможности QuickWin, можно создавать приложения с полноценным текстовым и графическим интерфейсом.

3. 2. Операции над окнами QUICKWIN.

3.2.1. Виды окон QUICKWIN.

После запуска приложения QuickWin создается *обрамляющее окно* (frame window). С обрамляющим окном в модуле MSFLIB связана целочисленная константа QWIN$FRAMEWINDOW, используемая в некоторых процедурах QuickWin, например в SETWSIZEQQ, для идентификации обрамляющего окна.

В обрамляющем окне могут быть расположены одно или несколько *дочерних окон* (child windows). Любое дочернее окно QuickWin появляется либо после вызова функции SETWINDOWCONFIG, либо после выполнения первой процедуры ввода/вывода (В/В), например подпрограммы CLEARSCREEN, заполняющей окно цветом фона. В приложениях QuickWin может быть до 40 дочерних окон. Причем в это число входят также дочерние окна, подсоединенные к устройствам, которые были закрыты оператором CLOSE. Такие окна недоступны для операции В/В данных.

3.2.2. Создание дочернего окна.

В общем случае дочернее окно создается оператором OPEN со спецификатором FILE=’USER’, например

```
open(100, file='user', title=' Изгиб балки ')
```

Спецификатор TITLE= является необязательным и используется для задания заголовка дочернего окна. Каждое новое дочернее окно подсоединяется к устройству В/В с уникальным номером, задаваемым спецификатором UNIT= (в данном случае можно было использовать и такую запись UNIT=100).

Однако оператор OPEN может в приложении отсутствовать. В этом случае в QuickWin создается одно дочернее окно, подсоединенное к всегда существующим устройствам 0,5 и 6. Создаваемое таким образом окно имеет заголовок Graphic 1.

3.2.3. Активизация дочернего окна.

Дочернее окно считается *активным*, если в него выполняется графический В/В, например процедурами LINETO_W, RECTANGLE_W и т.п.

3.2.4. Закрытие устройства дочернего окна.

Из программы устройство, к которому присоединено дочернее окно, закрывается оператором CLOSE. Дочернее окно можно оставить открытым (видимым) и после выполнения этого оператора, если в нем задать спецификатор STATUS='KEEP'. По умолчанию оператор CLOSE выполняется со спецификатором STATUS='DELETE', который может быть задан и явно, например:

```
close(2)                          ! Закрываем и устройство, и дочернее окно
close(unit=2, status='delete')    ! Закрываем и устройство, и дочернее окно
close(unit=2, status='keep')      ! Закрываем только устройство; дочернее окно
                                  !  открыто
```

3.2.5. Изменение размеров и позиции обрамляющего и дочернего окна.

Помимо параметров в QuickWin можно задать размеры и положение окна. Это выполняется функцией SETWSIZEQQ, имеющей синтаксис:

result4=SETWSIZEQQ(*unit*, *winfo*)

unit — выражение типа INTEGER(4), задающее устройство, к которому подсоединено окно. По умолчанию окно подсоединено к устройствам 0,5 и 6. Для установки размеров обрамляющего окна в качестве номера устройства следует использовать определенную в модуле MSFLIB именованную константу QWIN$FRAMEWINDOW.

winfo — переменная производного типа *qwinfo*, задающая физические координаты левого верхнего угла окна, его высоту и ширину. Производный тип *qwinfo* определен в модуле MSFLIB.

type (*qwinfo*) *winfo*;
winfo.type=1;
 или
winfo.type=2;
 или
winfo.type=3;
 или
winfo.type=4;

где 1 — это QWIN$MIN — выполняется минимизация окна;
2 — это QWIN$MAX — выполняется максимизация окна;
3 — это QWIN$RESTORE — минимизированное окно возвращается к прежним размерам и позиции;
4 — это QWIN$SET — положение окна и его размеры устанавливаются по значениям других компонентов *qwinfo*;

Функция возвращает значение типа INTEGER(4), равное нулю в случае успеха и отличное от нуля при неудаче.

Замечание. Позиция и размеры обрамляющего окна задаются в пикселях экрана. Позиция и размеры дочернего окна задаются в единицах высоты и ширины играфической литеры — знакоместа.

3. 3. Изменение системного меню.

По умолчанию приложение QuickWin сопровождается приведенным на рис.3.3. меню.

Рис.3.3.Меню QuickWin.

рассмотрим устройство меню. Оно состоит из пунктов; каждый пункт имеет подпункты, которые отображаются после воздействия (мышью или с клавиатуры) на пункт меню. Так пункт меню File имеет подпункты Print, Save, Exit (Ctrl+C). Пункты меню и подпункты имеют номера. Так, пункт File имеет номер 1, пункт Edit — номер 2 и т.д. Так же изменяются и номера подпунктов: подпункт Print имеет номер 1, Save — номер 2, а Exit — номер3. С каждым подпунктом меню связана установленная по умолчанию процедура, например с подпунктом Print связана процедура WINPRINT, запускаемая при выборе этого подпункта и открывающая диалоговое окно работы с принтером.

При инициализации можно создать свое собственное меню, которое в процессе работы программы может быть изменено. Также в процессе работы программы можно изменить и задаваемое по умолчанию меню. Функции, изменяющие меню и его свойства, приведены в табл.3.1. Все функции имеют тип LOGICAL(4).

Таблица 3.1.Функции, изменяющие меню и его свойства.

Функция	*Назначение*
APPENDMENUQQ	Добавляет пункт или подпункт меню
DELETEMENUQQ	Удаляет пункт или подпункт меню
INSERTMENUQQ	Вставляет пункт или подпункт меню
MODIFYMENUFLAGSQQ	Изменяет свойства пунктов или подпунктов меню, применяя одно из приведенных в табл.3.2. действий
MODIFYMENUROUTINEQQ	Изменяет связанную с пунктом меню или его подпунктом подпрограмму
MODIFYMENUSTRINGQQ	Изменяет название пункта или подпункта меню
SETWINDOWMENUQQ	Создает список имеющихся дочерних окон

Таблица 3.2.Изменение свойств пунктов и подпунктов меню.

Действие	*Результат*
$MENUGRAYED	Пункт или подпункт меню неактивен и светло-серого цвета
$MENUDISABLED	Пункт или подпункт меню неактивен, но обычного серого цвета
$MENUENABLED	Пункт или подпункт меню становится активным (доступным)
$MENUSEPARATOR	Вывод разделительной линии между подпунктами меню
$MENUCHECKED	Простановка галочки рядом с пунктом или подпунктом меню
$MENUUNCHECKED	Удаление галочки, стоящей рядом с пунктом или подпунктом меню

При необходимости можно в одной функции задать несколько действий, связав их логической операцией .OR., например:

$MENUCHECKED.OR. $MENUENABLED.

С каждым пунктом пользовательского меню можно связать либо заданную в QuickWin (табл.3.3.), либо пользовательскую подпрограмму, которая будет запускаться при выборе подпункта.

Таблица 3.3.Заданные процедуры QuickWin.

Имя процедуры	*Действие*
WINPRINT	Вывод содержимого окна на принтер
WINSAVE	Запись изображения в файл
WINEXIT	Выход из QuickWin
WINSELGRAPH	Выбор графического изображения из текущего окна
WINCOPY	Копирование выбранных элементов в буфер
WINCLEARPASTE	Очистка буфера
WINSIZETOFIT	Освобождение дочернего окна от передвижных шкал
WINFULLSCREEN	Переключение в режим полного экрана
WININDEX	Вывод содержания имеющейся справочной информации (Help)
WINUSING	Вывод информации о том, как использовать Help
NUL	Отсутствие процедуры

3. 4. Инициализация меню и обрамляющего окна.

Если программу QuickWin снабдить внешней функцией INITIALSETTINGS, то эта функция будет автоматически вызвана при запуске приложения. В этой функции можно задать начальные параметры обрамляющего окна (размеры и положение) и создать свое собственное меню взамен устанавливаемого по умолчанию меню QuickWin (см.рис.3.3.).

Функция INITIALSETTINGS может включать приведенные в табл.3.1. функции, которые изменяют меню QuickWin, и функцию SETWINSIZEQQ, задающую размеры и позицию обрамляющего окна QuickWin. Напомним, что параметры обрамляющего окна задаются в пикселях, а размеры и позиция дочернего окна — в единицах высоты и ширины знакоместа.

Пользовательское меню будет создано, если первой среди всех изменяющих меню функций выполнить функцию APPENDMENUQQ, добавляющую первый пункт меню, например:

l4 = appendmenuqq(1, $MENUENABLED,   'Файл'C, NUL)

Если INITIALSETTINGS не создает пользовательское меню, то будет загружено приведенное на рис.3.3. меню. Причем выполняемые в INITIALSETTINGS попытки удалить или вставить пункты (подпункты) меню будут проигнорированы.

Функция INITIALSETTINGS имеет тип LOGICAL(4) и должна вернуть .TRUE., если удалось выполнить начальные установки QuickWin, и .FALSE. — в противном случае.

Если функция INITIALSETTINGS в программе отсутствует, то QuickWin запускает свою функцию инициализации, которая всегда возвращает.TRUE..

3. 5. Вывод стандартного окна сообщений.

При необходимости, обратившись к функции MESSAGEBOXQQ, в приложении можно вывести стандартное окно сообщений. Функция имеет синтаксис

response=MESSAGEBOXQQ(*msg, caption, mtype*)

msg — выводимое сообщение; должно быть СИ-строкой.

caption — текст, выводимый в заголовке окна сообщений; должен быть СИ-строкой.

mtype — именованная константа, определяющая объекты (вид кнопок или икон) или атрибутов окна сообщений. Можно, использовать логическую операцию .OR., комбинировать несколько констант, например сочетание MB$YESNO.OR.MB$ICONQUESTION задает кнопки *Да* и *Нет* и икону с вопросительным знаком. Примеры констант *mtype* приведены в табл.3.4.

Таблица 3.4.Примеры констант mtype функции MESSAGEBOXQQ.

Именованная константа	*Выводимые объекты*
MB$ICONEXCLAMATION	Икона с восклицательным знаком
MB$ICONSTOP	Икона со знаком ×
MB$ICONQUESTION	Икона с вопросительным знаком
MB$YESNO	Кнопки *Да* и *Нет*
MB$YESNOCANCEL	Кнопки *Да*, *Нет* и *Отмена*

Отметим, что в случае русского Windows имена кнопок выводятся на русском языке (кроме названия кнопки OK).

3. 6. Копирование графики окна QUICKWIN.

В дочернем окне QuickWin можно выделить и скопировать в буфер присутствующее в нем графическое изображение. Для этого используются подпункты SelectGraph и CopyGraph пункта меню *График*. После выбора подпункта SelectGraph следует, используя мышь или Shift и стрелки клавиатуры, выделить график отмеченным прямоугольником.

Далее выбранная часть окна может быть скопирована в буфер (подпункт CopyGraph меню *График* или Ctrl+Ins). Затем содержимое буфера можно вставить, например, в создаваемый вами в среде Word документ или в стандартный редактор рисунков Paint Brash. Вставка производиться после выполнения Shift+Ins.

3. 7. Использование мыши.

3.7.1. Связанные с мышью события.

В QuickWin по умолчанию мышь используется для доступа к меню и управления обрамляющим и дочерними окнами (изменение их положения, размеров…). Все эти управляющие воздействия выполняются за пределами дочернего окна. Внутри дочернего окна мышью можно обеспечить выход из режима “Полный экран”, выделение графики и текста с целью их последующего копирования в буфер и переключение с одного дочернего окна на другое.

Однако QuickWin можно создать подпрограммы, которые фиксируют и соответствующим образом реагируют на те или иные происходящие с мышью события. Перечень таких событий приведен в табл.3.5.

Таблица 3.5.События, которые могут происходить с мышью.

Событие	*Описание*
MOUSE$LBUTTONDOWN	Нажата левая кнопка мыши
MOUSE$LBUTTONUP	Отпущена после нажатия левая кнопка мыши
MOUSE$LBUTTONDBLCLK	Двойной удар левой кнопкой мыши
MOUSE$RBUTTONDOWN	Нажата правая кнопка мыши
MOUSE$RBUTTONUP	Отпущена после нажатия правая кнопка мыши
MOUSE$RBUTTONDBLCLK	Двойной удар правой кнопкой мыши
MOUSE$MOVE	Перемещение мыши

Замечания:

1. Программы могут фиксировать лишь те события, которые произошли, когда мышь находилась в пределах дочернего окна.
2. При двойном ударе кнопкой мыши происходят 4 события: BUTTONDOWN, затем BUTTONUP, далее BUTTONDBLCLK и вновь BUTTONUP. Причем событие BUTTONDBLCLK происходит лишь в том случае, если временная разница между первым и вторым ударами мыши не превышает порогового значения, установленного для мыши в панели управления Windows. В противном случае повторно происходит событие BUTTONDOWN.
3. Перечисленные события задаются в виде именованных, определенных в модуле MSFLIB констант.

4. Программы, обрабатывающие происходящие с мышью события, имеют приоритет над двойным ударом мышью при выходе из режима “Полный экран”. Всегда можно покинуть этот режим, нажав Alt+Enter. В то же время при выборе текста и графики дочернего окна QuickWin эти подпрограммы имеют более низкий приоритет, чем процедуры выбора текста или графики.

3.7.2. Блокирующая функция WAITONMOUSEEVENT.

Блокирующая функция WAITONMOUSEEVENT приостанавливает выполнение программы до тех пор, пока не произойдет заданное связанное с мышью событие. Функция имеет синтаксис:

resul4=WAITONMOUSEEVENT (*mouseevents*, *keystate*, *x*, *y*)

mouseevents — события, задающиеся в виде именованных констант (см. табл.3.5.). Тип *mouseevents* — INTEGER(4).

keystate — этот параметр передает состояние кнопок мыши и клавиш Shift и Ctrl клавиатуры во время события. Параметр является комбинацией приведенных в табл.3.6. именованных констант. Комбинация составляется по средством логической операции .OR.

Таблица 3.6.Возможные константы, образующие параметр keystate.

Константа	*Описание*
MOUSE$KS_LBUTTON	Нажата левая кнопка мыши
MOUSE$KS_RBUTTON	Нажата правая кнопка мыши
MOUSE$KS_SHIFT	Нажата и удерживается во время события клавиша Shift клавиатуры
MOUSE$KS_CONTROL	Нажата и удерживается во время события клавиша Ctrl клавиатуры

x, *y* — *x* и *y* координаты мыши во время события.

Параметр *mouseevents* является входным (имеет вид связи IN), а параметры *keystate*, *x*, *y* — выходные, с видом связи OUT.

Функция возвращает значение типа INTEGER(4), равное в случае успеха именованной константе, описывающей заданное событие, или MOUSE$BADEVENT — константе обозначающей, что заданное событие не поддерживается.

Функция не возвращает значение до тех пор, пока не произойдет заданное событие.

4. Задача.

4. 1. Задание.

Создание интерфейса для определения исходных данных и наглядного представления результатов счета (графически) на примере задачи расчета балки на изгиб. Использовать приведенную выше теорию для реализации поставленной задачи.

В составленной программе должна вычисляться функция прогиба и выводиться на экран в виде рисунка балки в деформированном состоянии с прорисовкой опор и внешней нагрузки. На рисунке балка должна быть изображена так, чтобы максимальное значение прогиба не превышало $W_{max} = 0.1$, которое можно задать (нормировка).

4. 2. Расчетная схема.

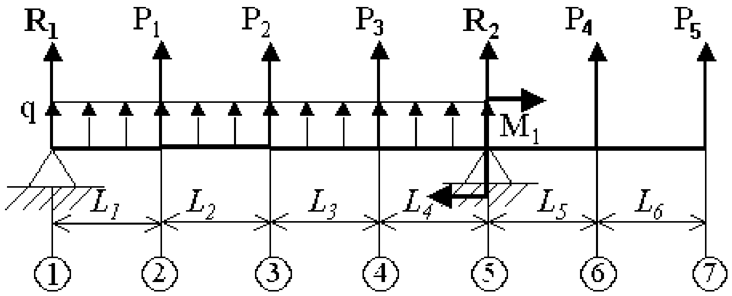


Рис.4.1. Расчетная схема.

4. 3. Исходные данные.

Значения силовых факторов и длины участков задаем числами в системе СИ (см. свой вариант).

Форму прогиба зададим выражением (универсальное уравнение прогиба):

$$W(x)=W_0+\theta_0 x+\frac{1}{EJ}\left\{R_1\frac{x^3}{6}+q\frac{x^4}{24}+P_1\frac{(x-L_1)^3}{6}+P_2\frac{(x-L_1-L_2)^3}{6}+P_3\frac{(x-L_1-L_2-L_3)^3}{6}+\right.$$

$$+R_2\frac{(x-L_1-L_2-L_3-L_4)^3}{6}-q\frac{(x-L_1-L_2-L_3-L_4)^4}{24}+M\frac{(x-L_1-L_2-L_3-L_4)^2}{2}+ \qquad (1)$$

$$\left.+P_4\frac{(x-L_1-L_2-L_3-L_4-L_5)^3}{6}\right\}$$

Из граничных условий $W(0)=W_0=0, \qquad W(L_1+L_2+L_3+L_4)=0$ (2)

Находим

$$0=W(L_1+L_2+L_3+L_4)=\theta_0(L_1+L_2+L_3+L_4)+\frac{1}{EJ}\left\{R_1\frac{(L_1+L_2+L_3+L_4)^3}{6}+\right.$$

$$\left.+q\frac{(L_1+L_2+L_3+L_4)^4}{24}+P_1\frac{(L_2+L_3+L_4)^3}{6}+P_2\frac{(L_3+L_4)^3}{6}+P_3\frac{(L_4)^3}{6}+P_4\frac{(-L_5)^3}{6}\right\}$$

$$\theta_0=\frac{1}{EJ(L_1+L_2+L_3+L_4)}\left\{R_1\frac{(L_1+L_2+L_3+L_4)^3}{6}+q\frac{(L_1+L_2+L_3+L_4)^4}{24}+\right.$$

$$\left.+P_1\frac{(L_2+L_3+L_4)^3}{6}+P_2\frac{(L_3+L_4)^3}{6}+P_3\frac{(L_4)^3}{6}+P_4\frac{(-L_5)^3}{6}\right\} \qquad (3)$$

Найдем коэффициент нормировки прогиба:
прогиб (1) представим через безразмерный прогиб

$$W(x)=\frac{B}{(L_1+L_2+L_3+L_4+L_5+L_6)}\overline{W} \qquad \Rightarrow \qquad B=\frac{W(x)\cdot(L_1+L_2+L_3+L_4+L_5+L_6)}{\overline{W}} \quad (4)$$

окончательно коэффициент нормировки B получим из условия максимального прогиба

$$B = \frac{W_{max} \cdot \left(L_1 + L_2 + L_3 + L_4 + L_5 + L_6\right)}{\overline{W}_{max}} = \frac{0.1 \cdot \left(L_1 + L_2 + L_3 + L_4 + L_5 + L_6\right)}{\overline{W}_{max}}, \quad (5)$$

где безразмерный максимальный прогиб ($\overline{W}_{max}$) ищется путем перебора значений по модулю.

В формулу для прогиба (1) входят неизвестные реакции опор R_1 и R_2, определим их из условий

$\sum M_{(1)} = 0$:

$$P_1L_1 + P_2\left(L_1 + L_2\right) + P_3\left(L_1 + L_2 + L_3\right) + R_2\left(L_1 + L_2 + L_3 + L_4\right) + P_4\left(L_1 + L_2 + L_3 + L_4 + L_5\right) +$$
$$+ P_5\left(L_1 + L_2 + L_3 + L_4 + L_5 + L_6\right) + q\frac{\left(L_1 + L_2 + L_3 + L_4\right)\cdot\left(L_1 + L_2 + L_3 + L_4\right)}{2} - M = 0$$

$$R_2 = \frac{1}{\left(L_1 + L_2 + L_3 + L_4\right)}\left\{- P_1L_1 - P_2\left(L_1 + L_2\right) - P_3\left(L_1 + L_2 + L_3\right) - P_4\left(L_1 + L_2 + L_3 + L_4 + L_5\right) - \right.$$
$$\left. - P_5\left(L_1 + L_2 + L_3 + L_4 + L_5 + L_6\right) - q\frac{\left(L_1 + L_2 + L_3 + L_4\right)\cdot\left(L_1 + L_2 + L_3 + L_4\right)}{2} + M\right\} \quad (6)$$

$\sum M_{(5)} = 0$:

$$P_4L_5 + P_5\left(L_5 + L_6\right) - P_1\left(L_2 + L_3 + L_4\right) - R_1\left(L_1 + L_2 + L_3 + L_4\right) - P_2\left(L_3 + L_4\right) +$$
$$- P_3L_4 - q\frac{\left(L_1 + L_2 + L_3 + L_4\right)\cdot\left(L_1 + L_2 + L_3 + L_4\right)}{2} - M = 0$$

$$R_1 = \frac{1}{\left(L_1 + L_2 + L_3 + L_4\right)}\left\{P_4L_5 + P_5\left(L_5 + L_6\right) - P_1\left(L_2 + L_3 + L_4\right) - P_2\left(L_3 + L_4\right) + \right.$$
$$\left. - P_3L_4 - q\frac{\left(L_1 + L_2 + L_3 + L_4\right)\cdot\left(L_1 + L_2 + L_3 + L_4\right)}{2} - M\right\} \quad (7)$$

4. 4. Структура оболочки.

После загрузки программы должно появиться окно программы со строкой меню. В качестве пунктов и подпунктов меню можно предложить:

Таблица 4.1. Вариант меню.

п/п	*Пункт*	*Подпункт*
1	Файл	Новый… Открыть… ________________ Сохранить Сохранить как… ________________ Печать ________________ Exit
2	Геометрия…	Диалоговое окно
3	Усилия…	Диалоговое окно
4	Настройка…	Диалоговое окно
5	Рисунок	Диалоговое окно
6	Вид	Полный экран (Full Screen)
7	График…	SelectGraph CopyGraph ________________ SaveWin
8	Help	Содержание... Как использовать Help...

9	О программе…	Диалоговое окно

4. 5. Структура программы.

4.5.1. Головная программа.

Головная программа используется для запуска проекта. В ней содержатся команды создания обрамляющего окна с именем “Изгиб балки” в шкале, максимизация размеров обрамляющего окна, цикл ожидания события (нажатие правой кнопки мыши).

4.5.2. Логическая функция создания меню.

В проект может быть включена функция создания собственного меню. Функция нигде явно не вызывается, но выполняется автоматически при запуске программы. Если при ее выполнении обнаруживаются ошибки, то загружается стандартное меню (см. рис.3.3.).

Все пункты меню описываются с помощью команды добавления пункта меню. Среди параметров команды можно выделить следующие:

- порядковый номер пункта или подпункта (подпункты имеют номер своего пункта);
- статус пункта;
- текст в виде СИ-строки — название пункта или подпункта;
- имя выполняемой подпрограммы, имеющей только один логический параметр для контроля выполнения. Подпрограммы могут быть стандартными (см. табл.3.3.) или подпрограммами пользователя. В последнем случае их имя должно быть указано в операторе EXTERNAL функции создания меню.

Все действия по настройке параметров задачи, счета и т.д. выполняются через пункты меню.

4.5.3. Обмен данными между программными единицами.

Такой обмен удобно выполнять с помощью модулей. В модуле описываются все глобальные объекты, которым можно задать начальные значения. Эти объекты становятся доступными под их именами после включения в программную единицу сразу после заголовка оператора

USE’имя модуля’.

4.5.4. Изменение исходных данных.

Исходные данные — параметры задачи и оболочки — могут меняться от запуска к запуску. Для сохранения последних введенных данных может использоваться текстовый файл, имеющий строго установленную структуру. Чтение данных из файла может выполняться или при запуске программы, или при выполнении соответствующего подпункта меню (Открыть…), или непосредственно перед изменением или использованием данных. Запись данных в файл может осуществляться или сразу после их редактирования, или по команде подпункта меню (Сохранить или Сохранить как…), или перед закрытием программы.

4.5.6. Редактирование данных.

Редактирование данных может быть осуществлено с помощью диалоговых окон, имеющих соответствующие поля. Диалоговое окно может быть раскрыто при выборе соответствующего пункта меню. При выполнении подпрограммы, установленной для такого пункта, выполняется инициализация диалогового окна, устанавливаются свойства полей и событий. Например, изменение какого-либо данного могут фиксироваться (программно) как в процессе редактирования, так и после закрытия диалогового окна. События могут выполняться при нажатии кнопок путем запуска подпрограмм, имеющих строго фиксированные параметры. Некоторые события устанавливаются по умолчанию.

Подробно как создаются диалоговые окно было рассмотрено ранее в главе 1.

Осталось нерассмотренным только диалоговое окно с рисунком, его можно создать так:

- создаем рисунок балке в среде Word (см.рис.4.1.) и сохраняем его в формате BMP.
- выполняем цепочку: Insert — Resource — Bitmap — Import… — выбираем свой файл BMP — Import
- выполняем цепочку: Insert — Resource — Dialog — вставляем Picture — нажать 2 раза левой кнопкой мыши на рисунок после чего появится окно

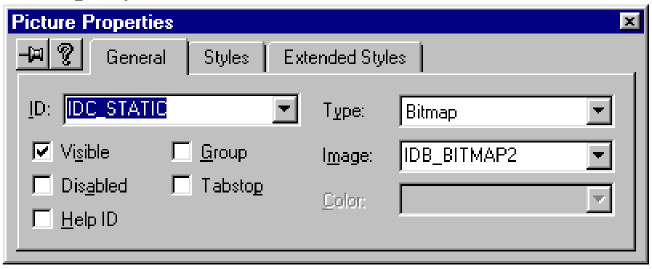


Рис.4.2.

4.5.7. Графическое представление результатов счета.

Данная подпрограмма содержит в себе команды создания графического (дочернего) окна с именем “График” в шкале, максимизация размеров графического окна. Далее для графического вывода используются функции, описанные подробно в главе 2. Затем вызов подпрограмм рисования стрелок, изображающих внешние нагрузки.

4. 6. Программа.

Имя проекта: BEAM

- File — New — Project Workspace — QuickWin Application — Name: BEAM и папку его размещения (Location) — Create.
- Головная программа: кнопка New Source File, правая кнопка мыши — Insert File into Project — BEAM — head1

```
$real:8
PROGRAM Iz_b_1_h
    USE MSFLIB
    implicit none
            type (qwinfo)qw;
            integer i4,ix,iy
    ! раскрытие обрамляющего окна с максимальными размерами:
      OPEN(100, file='user', title=' Изгиб балки ')
    qw.type=2; i4=setwsizeqq(qwin$framewindow,qw);
    i4=setwsizeqq(100,qw)
    ! выход из этога окна при помощи мышки (событие-нажата правая кнопка мыши):
      do while (.TRUE.)
            i4 = waitonmouseevent(MOUSE$RBUTTONDOWN, i4, ix, iy)
            if(.not.i4==MOUSE$BADEVENT)exit
      end do
    end
```

- Перед созданием меню необходимо разработать структуру файла TXT. В файле должны храниться:

  а)геометрические параметры:

  m_1, m_2, m_3, m_4, m_5, m_6 в формате f6.2

  б)числовые значения нагрузок жесткости и толщины балки:

  P1,P2,P3,P4,P5,MM1,q,EJ,tol в формате f10.4

  в)параметры построения графика:

относительный максимальный прогиб	wm в формате f6.2
координаты порта вывода рисунка в пикселях (левый угол и правый угол)	jpl,ipl,jpr в формате i5

цвета построения cvb,cvs,cvf,cvq,cvm в формате i4

- Записанным данным необходимо задать значения по умолчанию, причем они должны быть доступны из любой подпрограммы. Поэтому эти переменные обязательно описываем в модуле: New Source File, правая кнопка мыши — Insert File into Project — BEAM — mod_b

```
$real:8
Module dano
    ! длины участков нагружения
    real:: m1=1., m2=1., m3=1., m4=1., m5=1., m6=1.
    ! сумма некоторых длин участков нагружения
    real:: m123456, m1234
    ! реакции опор и параметр θ0
    real:: r1, r2, tet0
    ! внешняя нагрузка, жесткость балки и ее толщина
    real:: p1=1., p2=1.,p3=1.,p4=1.,p5=1., mm1=1., q=1., EJ=1., tol=1.
    ! относительный максимальный прогиб
    real:: wm=.1
    ! высоты опор и стрелок при их графической реализации
    real:: lp,lf,lm,lq
    ! нормирующий прогиб коэффициент
    real:: B
    integer:: jpl=100, ipl=100, jpr=400, ipr ,lam,laq
    integer:: cvb=0, cvs=5, cvf=15,cvq=12,cvm=3
    ! описание текстовых переменных
            character(6)::    fmt1='(f6.2)',c1
            character(10)::   fmt2='(f10.4)',c2
            character(5)::    fmt3='(i5)',c3
            character(4)::    fmt4='(i4)',c4
end module dano
```

- Меню Файл: с каждым подпунктом (см. табл.4.1.) связано выполнение подпрограммы.
  New_f:
  по оператору open(1,file='') открывается стандартное окно создания/открытия файла. В окне выбирается нужная папка, задается имя.txt, подтверждается создание нужного файла. Производится запись (call save_f(checked)) текущих исходных данных.
  В момент запуска программы доступны только подпункты: Новый…, Открыть…, Печать, Exit. После выполнения Новый… или Открыть… (определяется файл с логическим номером 1) становятся доступны все остальные подпункты меню.
  Open_f:
  open(1,file='') — выбирается нужный файл и производиться чтение из него.
  Save_f:
  запись в файл под номером 1 текущих данных
  Exit_pr:
  следует запрос на сохранение (если открыт файл №1). Если да, то запись. Затем Exit.

```
    $real:8
subroutine new_f(checked)
 use msflib
 logical checked
  checked=.true.
  open(1,file='');
  call save_f(checked)
  ! модификация пунктов меню
  do i=2,5;
        i4=modifymenuflagsqq(i,0,$menuenabled);
  enddo
    i4=modifymenuflagsqq(1,4,4)
    i4=modifymenuflagsqq(1,5,4)
        1 return
```

```
   end

subroutine save_f(checked)
  use dano
  logical checked
   checked=.true.
             rewind(1)
     write(1,'(6f6.2,12x,"m1,m2,m3,m4,m5,m6")')m1,m2,m3,m4,m5,m6
     write(1,'(9f10.4,3x, "P1,P2,P3,P4,P5,MM1,q,EJ,tol")')P1,P2,P3,P4,P5,MM1,q,EJ,tol
     write(1,'( f6.2,24x,"wm"      )')wm
     write(1,'( 3i5, 20x,"jpl,ipl,jpr"   )')jpl,ipl,jpr
     write(1,'( 5i4, 25x,"cvb,cvs,cvf,cvq"      )')cvb,cvs,cvf,cvq,cvm
end

subroutine open_f(checked)
 use msflib
 logical checked
   checked=.true.
   open(1,file='');
   call read_f
   ! модификация пунктов меню
              do i=2,5; i4=modifymenuflagsqq(i,0,$menuenabled);enddo
      i4=modifymenuflagsqq(1,4,4)
      i4=modifymenuflagsqq(1,5,4)
   end

subroutine read_f
  use dano
             rewind(1)
     read(1,*)m1,m2,m3,m4,m5,m6
     read(1,*)P1,P2,P3,P4,P5,MM1,q,EJ,tol
     read(1,*)wm
     read(1,*)jpl,ipl,jpr
     read(1,*)cvb,cvs,cvf,cvq,cvm
end

subroutine exit_pr(checked)
 use msflib
 logical  checked, exist_f
 integer i4
   inquire(1,named=exist_f)
   if(exist_f) then
      i4 = messageboxqq('Сохранить файл?'C,&
         '    BEAM 'C, MB$yesnocancel)
              if(i4==MB$IDYES ) call  save_f(checked)
              if(i4==MB$IDcancel ) return
            endif
   i4 =  SETEXITQQ(QWIN$EXITNOPERSIST)
   stop
end
```

- Меню Геометрия…: диалоговое окно

```
$real:8
$debug
! оформление диалогова окна "Геометрия...":
subroutine geometr(checked)
  use msflib
  use dialogm
  use dano
  include 'resource.fd'
```

```
  type (dialog) dlg
  logical checked, l4
  integer:: i4
    checked=.true.
    l4=dlginit(idd_geom,dlg)
        write(c1,fmt1)m1;     l4=dlgset(dlg,idc_dl_ed1,c1)
        write(c1,fmt1)m2;     l4=dlgset(dlg,idc_dl_ed2,c1)
        write(c1,fmt1)m3;     l4=dlgset(dlg,idc_dl_ed3,c1)
        write(c1,fmt1)m4;     l4=dlgset(dlg,idc_dl_ed4,c1)
        write(c1,fmt1)m5;     l4=dlgset(dlg,idc_dl_ed5,c1)
        write(c1,fmt1)m6;     l4=dlgset(dlg,idc_dl_ed6,c1)
        write(c2,fmt2)EJ;     l4=dlgset(dlg,idc_g_ed,c2)
        write(c2,fmt2)tol;    l4=dlgset(dlg,idc_t_ed,c2)
        write(c2,fmt2)wm;     l4=dlgset(dlg,idc_wm_ed,c2)

        i4=dlgmodal(dlg);       if(i4==idcancel)return

        l4=dlgget(dlg,idc_dl_ed1,c1); read(c1,fmt1)m1;
        l4=dlgget(dlg,idc_dl_ed2,c1); read(c1,fmt1)m2;
        l4=dlgget(dlg,idc_dl_ed3,c1); read(c1,fmt1)m3;
        l4=dlgget(dlg,idc_dl_ed4,c1); read(c1,fmt1)m4;
        l4=dlgget(dlg,idc_dl_ed5,c1); read(c1,fmt1)m5;
        l4=dlgget(dlg,idc_dl_ed6,c1); read(c1,fmt1)m6;
        l4=dlgget(dlg,idc_g_ed,c2);   read(c2,fmt2)EJ;
        l4=dlgget(dlg,idc_t_ed,c2);   read(c2,fmt2)tol;
        l4=dlgget(dlg,idc_wm_ed,c2);  read(c2,fmt2)wm;

      call dlgexit(dlg)
      call save_f(checked)
      Call coef
      Call gg
end
```

- Меню Усилия…: диалоговое окно

```
$real:8
$debug
! оформление диалогова окна "Усилия...":
subroutine stren(checked)
  use msflib
  use dialogm
  use dano
  include 'resource.fd'
  type (dialog) dlg
  logical checked, l4
  integer:: i4
    checked=.true.
    l4=dlginit(idd_stren,dlg)
        write(c2,fmt2)p1;     l4=dlgset(dlg,idc_s_ed1,c2)
        write(c2,fmt2)p2;     l4=dlgset(dlg,idc_s_ed2,c2)
        write(c2,fmt2)p3;     l4=dlgset(dlg,idc_s_ed3,c2)
        write(c2,fmt2)p4;     l4=dlgset(dlg,idc_s_ed4,c2)
        write(c2,fmt2)p5;     l4=dlgset(dlg,idc_s_ed5,c2)
        write(c2,fmt2)mm1;    l4=dlgset(dlg,idc_m_ed,c2)
        write(c2,fmt2)q;      l4=dlgset(dlg,idc_q_ed,c2)

        i4=dlgmodal(dlg);       if(i4==idcancel)return

        l4=dlgget(dlg,idc_s_ed1,c2);  read(c2,fmt2)p1;
        l4=dlgget(dlg,idc_s_ed2,c2);  read(c2,fmt2)p2;
        l4=dlgget(dlg,idc_s_ed3,c2);  read(c2,fmt2)p3;
```

```
        l4=dlgget(dlg,idc_s_ed4,c2);   read(c2,fmt2)p4;
        l4=dlgget(dlg,idc_s_ed5,c2);   read(c2,fmt2)p5;
        l4=dlgget(dlg,idc_m_ed,c2);    read(c2,fmt2)mm1;
        l4=dlgget(dlg,idc_q_ed,c2);    read(c2,fmt2)q;

      call dlgexit(dlg)
      call save_f(checked)
      Call coef
      Call gg
end
```

- Меню Настройка…: диалоговое окно

```
$real:8
! оформление диалогова окна "Настройка...":
subroutine tuning(checked)
  use msflib
  use dialogm
  use dano
  include 'resource.fd'
  type (dialog) dlg
  logical checked, l4
  integer:: i4
   checked=.true.
   l4=dlginit(idd_tun,dlg)

        write(c4,fmt4)cvb;     l4=dlgset(dlg,idc_col1_ed,c4)
        write(c4,fmt4)cvs;     l4=dlgset(dlg,idc_col2_ed,c4)
        write(c4,fmt4)cvf;     l4=dlgset(dlg,idc_col3_ed,c4)
        write(c3,fmt3)jpl;     l4=dlgset(dlg,idc_par2_ed,c3)
        write(c3,fmt3)ipl;     l4=dlgset(dlg,idc_par3_ed,c3)
        write(c3,fmt3)jpr;     l4=dlgset(dlg,idc_par4_ed,c3)
        write(c4,fmt4)cvq;     l4=dlgset(dlg,idc_col4_ed,c4)
        write(c4,fmt4)cvm;     l4=dlgset(dlg,idc_col5_ed,c4)

        i4=dlgmodal(dlg);        if(i4==idcancel)return

        l4=dlgget(dlg,idc_col1_ed,c4);          read(c4,fmt4)cvb;
        l4=dlgget(dlg,idc_col2_ed,c4);          read(c4,fmt4)cvs;
        l4=dlgget(dlg,idc_col3_ed,c4);          read(c4,fmt4)cvf;
        l4=dlgget(dlg,idc_par2_ed,c3);          read(c3,fmt3)jpl;
        l4=dlgget(dlg,idc_par3_ed,c3);          read(c3,fmt3)ipl;
        l4=dlgget(dlg,idc_par4_ed,c3);          read(c3,fmt3)jpr;
        l4=dlgget(dlg,idc_col4_ed,c4);          read(c4,fmt4)cvq;
        l4=dlgget(dlg,idc_col5_ed,c4);          read(c4,fmt4)cvm;

      call dlgexit(dlg);
      call save_f(checked)
      Call coef;
      Call gg
end
```

<u>Замечание.</u> Здесь после нажатия на кнопку OK любого диалогового окна происходит пересчет прогиба и его построение заново. Для этого вызываем подпрограмму построения прогиба   Call gg.

- Меню Рисунок: диалоговое окно

```
$real:8
! оформление диалогова окна "Рисунок":
subroutine ris(checked)
  use msflib
  use dialogm
```

```
  use dano
  include 'resource.fd'
  type (dialog) dlg
  logical checked, l4
  integer:: i4
    checked=.true.
    l4=dlginit(idd_ris,dlg)
            i4=dlgmodal(dlg)
          call dlgexit(dlg)
end
```

- Меню Вид, График и Help: см. табл.3.3. — заданные процедуры QuickWin.
- Меню О программе…: диалоговое окно.

```
$real:8
! оформление диалогова окна "Изгиб балки !":
subroutine wind(checked)
  use msflib
  use dialogm
  use dano
  include 'resource.fd'
  type (dialog) dlg
  logical checked, l4
  integer:: i4
    checked=.true.
    l4=dlginit(idd_dp,dlg)
            i4=dlgmodal(dlg)
          call dlgexit(dlg)
end
```

- Подпрограмма инициализации меню:

```
! инициализация меню:
LOGICAL(4) FUNCTION InitialSettings()

USE MSFLIB
implicit none
          LOGICAL(4)  l4
          EXTERNAL  new_f, open_f, save_f, exit_pr, geometr, stren, tuning, ris, wind

l4 = appendmenuqq(1, $MENUENABLED,   'Файл'C,           NUL     )
l4 = appendmenuqq(1, $MENUENABLED,   'Новый...'C,        new_f   )
l4 = appendmenuqq(1, $MENUENABLED,   'Открыть...'C,      open_f  )
l4 = appendmenuqq(1, $MENUSEPARATOR, 'sep'C,             NUL     )
l4 = appendmenuqq(1, $MENUGRAYED,    'Сохранить'C,       save_f  )
l4 = appendmenuqq(1, $MENUGRAYED,    'Сохранить как...'C, new_f   )
l4 = appendmenuqq(1, $MENUSEPARATOR, 'sep'C,             NUL     )
l4 = appendmenuqq(1, $MENUENABLED,   'Печать'C,          WINPRINT    )
l4 = appendmenuqq(1, $MENUSEPARATOR, 'sep'C,             NUL     )
l4 = appendmenuqq(1, $MENUENABLED,   'Exit'C,            exit_pr  )

l4 = appendmenuqq(2, $MENUgrayed,    'Геометрия...'C,    geometr   )

l4 = appendmenuqq(3, $MENUgrayed,   'Усилия...'C,        stren  )

l4 = appendmenuqq(4, $MENUENABLED,   'Настройка...'C,       tuning  )
l4 = appendmenuqq(5, $MENUENABLED,   'Рисунок'C,         ris  )

l4 = appendmenuqq(6, $MENUENABLED,    'Вид'C,             NUL     )
l4 = appendmenuqq(6, $MENUENABLED,   'Full Screen'C,  WINFULLSCREEN)

l4 = appendmenuqq(7, $MENUENABLED,    'График...'C,        nul  )
```

```
l4 = appendmenuqq(7, $MENUENABLED,   'SelectGraph'C, WINselectgraphics)
l4 = appendmenuqq(7, $MENUENABLED,   'CopyGraph'C, WINcopy)
l4 = appendmenuqq(7, $MENUSEPARATOR, 'sep'C,              NUL      )
l4 = appendmenuqq(7, $MENUENABLED,   'SaveWin'C, WINsave)


l4 = appendmenuqq(8, $MENUENABLED,   'Help'C,           NUL)
l4 = appendmenuqq(8, $MENUENABLED,   'Содержание...'C, WININDEX)
l4 = appendmenuqq(8, $MENUENABLED,   'Как использовать Help...'C, WINUSING)

l4 = appendmenuqq(9, $MENUENABLED,   'О программе...'C, wind)

            InitialSettings = l4
return
end
```

- Подпрограмма вычислений:

```
! вычисление реакций опрор (R1,R2) и начального угла поворота (tet0):
Subroutine coef
  use dano
   real::m12,m123,m12345,m56,m234,m34
   real::f,t,ft
   real(8)::W
                        m123456=m1+m2+m3+m4+m5+m6
                        m1234=m1+m2+m3+m4
                        m12=m1+m2;   m123=m1+m2+m3;       m12345=m1234+m5
                        m56=m5+m6;   m234=m1234-m1;       m34=m3+m4
r1=(p4*m5+p5*m56-mm1-p1*m234-p2*m34-p3*m4-0.5*q*m1234**2)/m1234
r2=(mm1-0.5*q*m1234**2-p1*m1-p2*m12-p3*m123-p4*m12345-p5*m123456)/m1234
tet0=-(r1*m1234**3/6.+q*m1234**4/24.+p1*m234**3/6.+p2*m34**3/6.+&
                        p3*m4**3/6.+p4*(-m5)**3/6.)/(m1234*EJ)

!поиск максимального значения прогиба:
    f=0;B=1
             do t=0.,m123456,.01
                        ft=dabs(W(dfloat(t))); if(ft>f)f=ft
             enddo
! нормировка прогиба:
    B=Wm/f*(m1+m2+m3+m4+m5+m6)
end

! запись функции прогиба (W):
real(8) function W(t)
  use dano
  real(8)::t
                     W=tet0*t+(r1*t**3/6.+q*t**4/24.)/EJ
            if(t>=m1)                                W=W+p1*(t-m1)**3/(EJ*6.)
            if(t>=m1+m2)                    W=W+p2*(t-m1-m2)**3/(EJ*6.)
            if(t>=m1+m2+m3)                 W=W+p3*(t-m1-m2-m3)**3/(EJ*6.)
            if(t>=m1+m2+m3+m4)              W=W+(r2*(t-m1-m2-m3-m4)**3/6.-&
                                                   q*(t-m1-m2-m3-m4)**4/24.+&
                                                   mm1*(t-m1-m2-m3-m4)**2/2.)/EJ
    if(t>=m1+m2+m3+m4+m5)  W=W+p4*(t-m1-m2-m3-m4-m5)**3/(EJ*6.)
   W=B*W
end

! запись функции угла поворота (alfa):
real(8) function alfa(t)
use dano
real(8)::t
```

```
            alfa=tet0+(r1*t**2/2.+q*t**3/6.)/EJ
         if(t>=m1)                          alfa=alfa+p1*(t-m1)**2/(EJ*2.)
         if(t>=m1+m2)                alfa=alfa+p2*(t-m1-m2)**2/(EJ*2.)
         if(t>=m1+m2+m3)             alfa=alfa+p3*(t-m1-m2-m3)**2/(EJ*2.)
         if(t>=m1+m2+m3+m4)          alfa=alfa+(r2*(t-m1-m2-m3-m4)**2/2.-&
                                              q*(t-m1-m2-m3-m4)**3/6.+&
                                              mm1*(t-m1-m2-m3-m4)/1.)/EJ
    if(t>=m1+m2+m3+m4+m5)  alfa=alfa+p4*(t-m1-m2-m3-m4-m5)**2/(EJ*2.)
  alfa=b*alfa
  alfa=datan(alfa)
end
```

- Подпрограммы графических построений:

```
$real:8
$debug
! посторение графика прогиба:
subroutine gg
  use msflib
  use dano
        integer,parameter::m=101
        integer(4)::i4
        real(8):: lxp,lyp ! высота в пикселях по x и по y
        real(8):: x,y,al
        real(8):: dx,dy,w,so
! задание переменных производного типа:
        type (qwinfo)qw; type(wxycoord)wxy,pxy(3)
! задание высот окна, опор,стрелок:
              lp=1;  lf=0.05*m123456;lq=0.02*m123456; lm=0.05*m123456
! длины:
     m123456=m1+m2+m3+m4+m5+m6;        m1234=m1+m2+m3+m4
! дочернего окна, где рисуется график (задается его конфигурация):
  close(2)
  ! название этого окна:
        OPEN(2, file='user', title=' График ')
  ! задаем размеры (распахиваем максимально) и позицию этого окна:
        qw.type=2;  i4=setwsizeqq(100,qw)
  ! задаем цвет экрана:
        ii=setbkcolor(cvf)
  ! открываем видовой порт (видовую систему координат):
        CALL SETVIEWPORT (0, 0, 800, 600)
  ! отчистка и заполнение экрана текущим цветом (cvf):
        CALL CLEARSCREEN ($GCLEARSCREEN)
  ! вычислем размеры видео окна:
              dx=1.5*m123456
              dy=2*(wm*m123456+lp)
              lxp=jpl+jpr;      lyp=lxp*dy/dx;  ipr=ipl+lyp
  ! открываем видовой порт (по размерам графика):
        CALL SETVIEWPORT (jpl, ipl, jpr, ipr)
  ! задаем цвет текущий цвет рисования:
        ii=setcolor(cvb)
  ! открываем окно вывода, в котором можем работатьс вещ. координатами:
        ii=setwindow(.true.,-5d-2*m123456,dy/2.,1.1d0*m123456,-dy/2.)
  ! граница окно вывода (обрамляем прямоугольником):
        ii= rectangle_w(2,-5d-2*m123456,dy/2.,1.1d0*m123456,-dy/2.)
  ! задаем текущую позицию графического вывода, т.е. ставим карандаш в точку начала рисования
        CALL MOVETO_W (0d0, W(0d0),wxy )
! рисуем среднюю линию балки:
                    ! шаг рисования
                    h=0.01*m123456
                    do i=1,m
```

```
                    ! вычисляем точки
                        x=i*h; al=alfa(x)
                        y=W(x)
                    ! соединям найденные точки линией
                        ii=lineto_w(x,y)
                    end do
! рисуем верхнюю линию балки:
                    al=alfa(0d0); y=W(0d0)+tol/2.*dcos(al)
                    CALL MOVETO_W (-tol/2.*dsin(al), y,wxy )
                    do i=1,m
                        x=i*h; al=alfa(x);
                        y=W(x)+tol/2.*dcos(al)
                        ii=lineto_w(x-tol/2.*dsin(al),y)
                    end do
! рисуем правый торец балки:
                ii=lineto_w(x+tol/2.*dsin(al),y-2.*tol/2.*dcos(al))
! рисуем нижнюю линию балки:
                    do i=m-1,0,-1
                        x=i*h; al=alfa(x)
                        y=W(x)-tol/2.*dcos(al)
                        ii=lineto_w(x+tol/2.*dsin(al),y)
                    end do
! рисуем левый торец балки:
                    ii=lineto_w(x-tol/2.*dsin(al),y+2.*tol/2.*dcos(al))
! рисуем опоры:
                    ! задаем ширину опоры
                     so=.05*m123456
                    ! неподвижная опора
                        x=0; al=alfa(x); y=W(x)-tol/2.*dcos(al)
                        x=x+tol/2.*dsin(al)
                            pxy.wx=(/x,x+so/2,  x-so/2/)
                            pxy.wy=(/y,y-.85*so,y-.85*so/)
                        ii=polygon_w(2,pxy,3)
                    ! "земля" под опорой
                    call moveto_w(x-0.75*so,y-0.85*so,wxy);      ii=lineto_w(x+0.75*so,y-0.85*so)
                    call moveto_w(x-0.5*so,y-0.85*so,wxy);       ii=lineto_w(x-0.75*so,y-1.05*so)
                    call moveto_w(x-0.25*so,y-0.85*so,wxy);      ii=lineto_w(x-0.5*so,y-1.05*so)
                    call moveto_w(x,y-0.85*so,wxy);              ii=lineto_w(x-0.25*so,y-1.05*so)
                    call moveto_w(x+0.25*so,y-0.85*so,wxy);      ii=lineto_w(x,y-1.05*so)
                    call moveto_w(x+0.5*so,y-0.85*so,wxy);       ii=lineto_w(x+0.25*so,y-1.05*so)

                    ! подвижная опора
                        x=m1234; al=alfa(x); y=W(x)-tol/2.*dcos(al)
                        x=x+tol/2.*dsin(al)
                            ii=ellipse_w(2,x-0.1*so,y,x+0.1*so,y-0.2*so)
                            ii=ellipse_w(2,x-0.1*so,y-0.65*so,x+0.1*so,y-0.85*so)
                        CALL MOVETO_W (x, y-0.2*so,wxy )
                      ii=lineto_w(x,y-0.65*so)
                    ! "земля" под опорой
                    call moveto_w(x-0.75*so,y-0.85*so,wxy);      ii=lineto_w(x+0.75*so,y-0.85*so)
                    call moveto_w(x-0.5*so,y-0.85*so,wxy);       ii=lineto_w(x-0.75*so,y-1.05*so)
                    call moveto_w(x-0.25*so,y-0.85*so,wxy);      ii=lineto_w(x-0.5*so,y-1.05*so)
                    call moveto_w(x,y-0.85*so,wxy);              ii=lineto_w(x-0.25*so,y-1.05*so)
                    call moveto_w(x+0.25*so,y-0.85*so,wxy);      ii=lineto_w(x,y-1.05*so)
                    call moveto_w(x+0.5*so,y-0.85*so,wxy);       ii=lineto_w(x+0.25*so,y-1.05*so)
! рисуем стрелки:
! распределенная нагрузка
                        do x=0.,m1234,0.1*m1234;if(q/=0)call strela_q(x);enddo
                    ! момент
```

```
                    if(mm1/=0) call  mom()
              ! сосредоточенные силы:
                    if(p1/=0) call strela_f(m1,p1);
                    if(p2/=0) call strela_f(m1+m2,p2)
                    if(p3/=0) call strela_f(m1+m2+m3,p3);
                    if(p4/=0) call strela_f(m1234+m5,p4)
                    if(p5/=0) call strela_f(m1234+m5+m6,p5)
end
```

Подпрограммы рисования стрелок (учтено направление сил и моментов, относительно исходной расчетной схемы, где приняты положительные направления):

```
$debug
$real:8
subroutine mom()
use msflib
use dano
      real(8)::x
      real(8)::al
            type(wxycoord)trxy(3)
            ii=setcolor(cvm)
            lam=1;if(mm1<0)lam=-1
        x=m1234;  al=alfa(x); y=W(x)+tol/2.*dcos(al)
             x=x-tol/2.*dsin(al)
             trxy.wx=(/x+lam*(2*lm/3.-0.05*lm),x+lam*(lm-0.05*lm),  x+lam*(2*lm/3.-0.05*lm)/)
             trxy.wy=(/y+lm,y+0.85*lm,y+0.7*lm/)
             ii=polygon_w(3,trxy,3)
             ii=rectangle_w(3,x-lam*5d-2*lm,y+9d-1*lm,x+lam*(2*lm/3.-5d-2*lm),y+8d-1*lm)
                   ii=rectangle_w(3,x-5d-2*lm,y+8d-1*lm,x+5d-2*lm,y)
end

subroutine strela_f(x1,p)
  use msflib
  use dano
            real(8)::al,x,w
        type(wxycoord)wxy(3)
                  lap=1;if(p<0)lap=-1
        ii=setcolor(cvs);x=x1
        al=alfa(x); y=W(x)+tol/2.*dcos(al)
                   x=x-tol/2.*dsin(al)
                   wxy.wx=(/x-0.2*lf, x+0.2*lf, x/)
                   wxy.wy=(/y+lap*2*lf/3., y+lap*2*lf/3., y+lap*lf/)
                   ii=polygon_w(3,wxy,3)
                   ii=rectangle_w(3,x-5d-2*lf,y+lap*2*lf/3.,x+5d-2*lf,y)
end

subroutine strela_q(x1)
  use msflib
  use dano
            real(8)::al,x
        type(wxycoord)wxy(3)
        ii=setcolor(cvq);x=x1
                  laq=1;if(q<0)laq=-1
            al=alfa(x); y=W(x)+tol/2.*dcos(al)
                   x=x-tol/2.*dsin(al)
                   wxy.wx=(/x-0.2*lq, x+0.2*lq, x/)
                   wxy.wy=(/y+laq*2*lq/3.,y+laq*2*lq/3.,y+laq*lq/)
                   ii=polygon_w(3,wxy,3)
                   ii=rectangle_w(3,x-5d-2*lq,y+laq*2*lq/3.,x+5d-2*lq,y)
end
```

4. 7. Графический результат счета.

Здесь

$$L_1 = L_2 = L_3 = L_4 = L_5 = L_6 = 1м$$
$$P_1 = P_2 = P_3 = P_4 = P_5 = 1H$$
$$h = 0.2м - толщина \quad балки$$
$$w_{\max} = 0.1$$
$$EJ = 1Нм^2$$
$$q = 1\,H/м$$
$$M = 1Нм$$

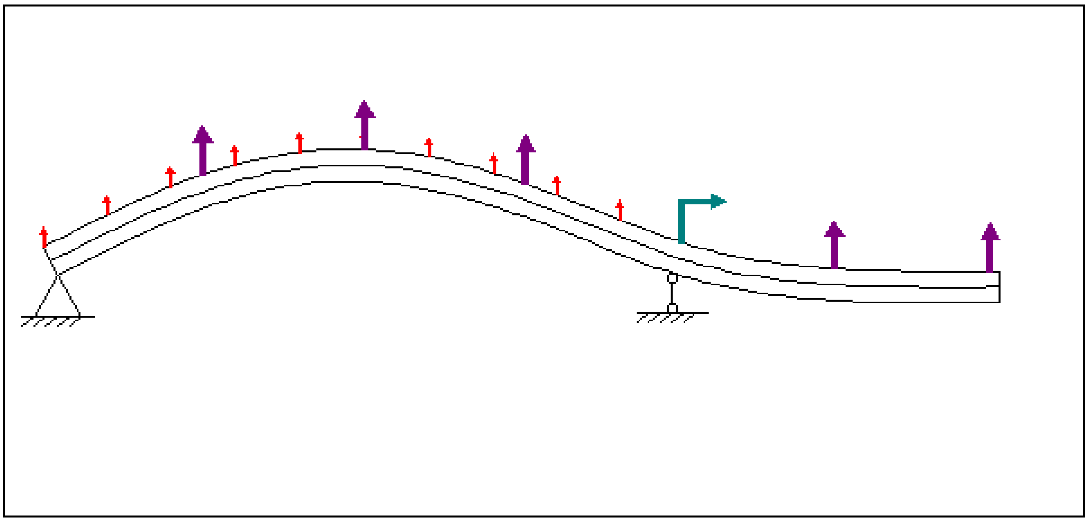


Рис.4.3.Прогиб балки.

Комплект заданий для зачета

по дисциплине *Практикум по алгоритмизации и программированию*
(наименование дисциплины)

Задача (задание) 1.Таблица футбольного чемпионата задана вектором X(10) (X(i)=i, где i- исходный номер команды) и матрицей A(10,10), где элемент A(i,j) численно равен количеству мячей, забитых i-ой командой j-ой. Постройте диаграмму результативности (количества забитых мячей) команд в зависимости от занятого места. Распечатайте таблицу распределения мест, набранных очков и результативность команд. Исходную матрицу А задайте самостоятельно.

Задача (задание) 2.Таблица футбольного чемпионата задана вектором X(10) (X(i)=i, где i- исходный номер команды) и матрицей A(10,10), где элемент A(i,j) численно равен количеству мячей, забитых i-ой командой j-ой. Найдите номера команд, имеющих больше побед, чем поражений. Постройте диаграмму результативности (количества забитых мячей) этих. Распечатайте эти команды, набранные ими очки и количество побед. Исходную матрицу А задайте самостоятельно.

Задача (задание) 3.Таблица футбольного чемпионата задана матрицей A(10,10) (A(i,i)=0, A(i,j)={0;1;2},i<>j) и вектором X(10) (X(i)=i, где i- исходный номер команды). Найдите номера команд, прошедших чемпионат без поражений. Постройте диаграмму результативности (количества забитых мячей) этих команд. Распечатайте эти команды, набранные ими очки и количество побед. Исходную матрицу А задайте самостоятельно.

Задача (задание) 4.Для целочисленной матрицы A(10,10) найдите матрицу из нулей и единиц B(10,10), элементы которой B(i,j) равны 1, если все соседи (элементы матрицы A, индексы которых отличаются от i,j не более чем на 1) меньше самого A(i,j). Организуйте графический вывод (в виде таблиц) результатов работы программы на экран. Исходную матрицу A задайте самостоятельно.

Задача (задание) 5.Для целочисленной матрицы A(10,10) найдите матрицу из нулей и единиц B(10,10), элементы которой B(i,j) равны 1, если среди соседей A(i,j) (элементов матрицы A, индексы которых отличаются от i,j не более чем на 1) есть не менее двух совпадающих с A(i,j). Организуйте графический вывод (в виде таблиц) результатов работы программы на экран. Исходную матрицу A задайте самостоятельно.

Задача (задание) 6.Дан целый массив A(20).Получите массив B(m) (динамический) из чисел,взятых по одному из каждой группы равных членов массива A. Постройте диаграмму количества одинаковых элементов в зависимости от их численных значений. Исходный массив A задайте самостоятельно.

Задача (задание) 7.Дан целый массив A(20). Получите массив B(m) (динамический), элементы которого равны всем целым в порядке возрастания из интервала [min(A(i)),max(A(i))], не входящим в А. Постройте диаграмму зависимости численных значений элементов массива B от порядкового номера. Исходный массив A задайте самостоятельно.

Задача (задание) 8.Найдите первые три корня a(i), i=1,2,3 нелинейного уравнения ch(a)*cos(a)+1=0.

Постройте графики трех функций

y(i)=U(a(i)*x) - S(a(i))/T(a(i))*V(a(i)*x), i=1,2,3, 0<=x<=1,

где S(x),T(x),U(x),V(x)- функции Крылова

Задача (задание) 9.Постройте графики трех функций Y(i,x)=C(i)*W(i,x), i=1,2,3,

где: W(i,x)=U(a(i)*x)-S(a(i))/T(a(i))*V(a(i)*x), 0<=x<=1,

a(1)=1.875, a(2)=4.694, a(3)=5*pi/2

C(i)=sqrt(1/(W(i,x)**2*dx)),

S(x),T(x),U(x),V(x)- функции Крылова.

Задача (задание) 10.Найдите методом простых итераций min собственное значение d и соответствующий собственный вектор для системы линейных однородных уравнений X=d*H*X

где

$$H = \begin{Vmatrix} c1+c2+c5 & -c2 & -c5 \\ -c2 & c2+c3 & -c3 \\ -c5 & -c3 & c3+c4+c5 \end{Vmatrix}$$

Постройте диаграмму зависимости численных значений элементов собственного вектора от порядкового номера.

Вычисления выполните для следующих значений: c1=1, c2=1.8, c3=0.8, c4=2, c5=.5

Задача (задание) 11.Используя метод наименьших квадратов заданную табличную функцию

x:	-.5	1	2	3	4	5	6
y:	1	.2	0	.2	.5	1.3	2

аппроксимируйте многочленом третей степени Q(x)=q0+q1*x+q2*x2+q3*x3. На печать выдайте значения qi и Q(x) в заданных точках xi. Постройте график многочлена и на нем покажите заданные точки.

Задача (задание) 12.Дана функция комплексного переменного

Графически показать, в какую фигуру в плоскости z преобразуется окружность

Исследовать случаи n=1,2,10,100

Задача (задание) 13.Для области D в плоскости z=x+i*y D={ 0<arg(z)<pi/2, 1<|z|<2 } постройте область D* в плоскости w=u+i*v, отображенную функцией w=ln(i*z) (область D при обходе должна оставаться слева). Результат представьте в виде графиков.

Теоретические вопросы

1. Решение краевой задачи для дифференциального уравнения второго порядка методом прогонки

2. Аппроксимация функций

3. Машинная графика для построения эпюр

4. Методы решения задач на собственные значения

5. Математическая модель и программа решения задачи по теории колебаний.

6. Комплексные величины при выполнении расчетов

Критерии оценки

- Задание считается выполненным на **пороговом** уровне, если студент ориентируется в теоретический вопросах, оценка составляет 50 баллов
- Задание считается выполненным на **базовом** уровне, если студент отвечает на теоретический вопрос и допускает незначительную ошибку при решении задачи, оценка составляет 75 баллов
- Задание считается выполненным на **продвинутом** уровне, если студент отвечает на теоретический вопрос и решает задачу, оценка составляет 100 баллов

Зачет считается сданным, если средняя сумма баллов по всем заданиям составляет не менее 50 баллов (по 100 балльной шкале).
Коэффициент, с которым учитывается полученная сумма баллов в общей оценке по дисциплине, определяется Правилами аттестации.

Составитель ________________________ А.И. Белоусов
(подпись)

«____»________________20 г.

ЦЕНТРАЛЬНОЕ РАСТЯЖЕНИЕ И СЖАТИЕ ПРЯМОГО БРУСА.

ЗАДАНИЕ: вычислить таблицы значений и построить эпюры продольных сил N, нормальных напряжений σ и перемещений u поперечных сечений прямого бруса (рис. 1).

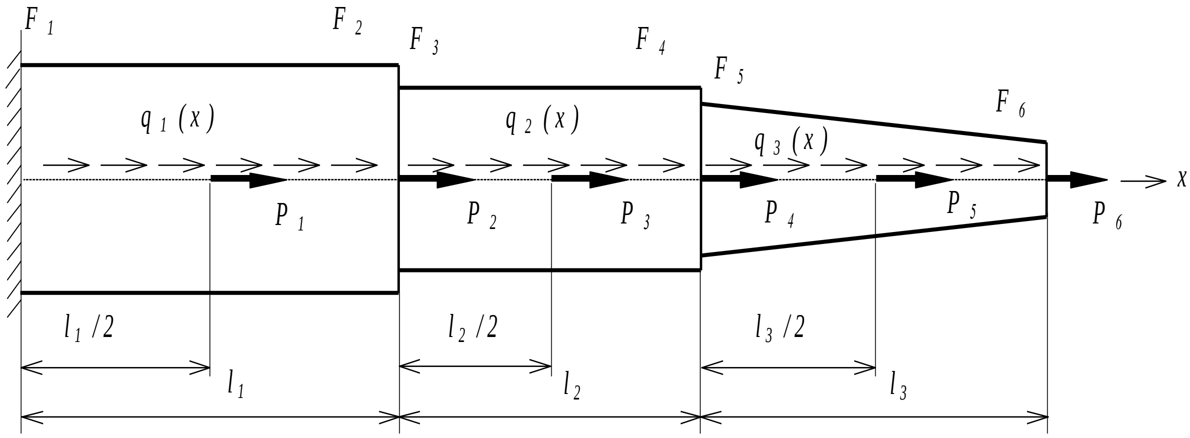


Рис. 1

Геометрически брус можно разделить на три участка. На i-ом участке (i =1, 2, 3) длиной l_i заданы F_{2i+1} , F_{2i} - площади левого и правого сечений участка бруса (на каждом участке принят линейный закон измененения площади по длине); P_{2i+1} , P_{2i} - сосредоточенные силы, приложенные в соответствии со схемой (рис. 1); $q_i(x)$ - продольная распределенная нагрузка, изменяющаяся по линейному закону (заданы k_i - тангенс наклона в уравнении прямой и b_i - значение q_i на левой границе участка).

Для вычисления значений N, σ, u необходимо воспользоваться следующими соотношениями:

$$N(x) = \int_x^{l_1+l_2+l_3} q(x)dx + \sum P_j \qquad (1)$$

В сумме (1) учитываются все сосредоточенные силы, приложенные на участке от x до $l_1+l_2+l_3$.

$$\sigma(x) = N(x) / F(x) \qquad (2)$$

$$u\,(x) = \frac{1}{E}\int_0^x \sigma\,(x)d\,x \qquad (3)$$

Для решения задачи требуется разбить брус на m силовых участков (в общем случае m = 6) и на каждом участке силовом вычислить требуемые величины. Решение представить в виде таблицы и эпюр. На каждом силовом участке рассчитать не менее трех точек.

Для определения варианта задания порядковый номер по групповому журналу перевести в двоичный код и дополнить нулями до 6-ти позиций (например №3=000011). В зависимости от значений в каждой позиции (0 или 1) по таблице определить исходные данные. (E=1.1·10^8 кН / м^2)

№ позиции	1-ая			2-ая			3-ья		
значения	l_1	$F_1 \cdot 10^4$	$F_2 \cdot 10^4$	l_2	$F_3 \cdot 10^4$	$F_4 \cdot 10^4$	l_3	$F_5 \cdot 10^4$	$F_6 \cdot 10^4$
в позиции	м	$м^2$	$м^2$	м	$м^2$	$м^2$	м	$м^2$	$м^2$
0	0.5	5	3	0.4	4	3	0.5	3	3
1	0.3	4	4	0.7	5	3	0.4	4	2

№ позиции	4-ая				5-ая				6-ая			
значения	P_1	P_2	k_1	b_1	P_3	P_4	k_2	b_2	P_5	P_6	k_3	b_3
в позиции	кН	кН	кН/м^2	кН/м	кН	кН	кН/м^2	кН/м	кН	кН	кН/м^2	кН/м
0	0	10	5	2	0	-5	-4	-4	0	20	-5	-3
1	10	-10	0	0	-10	20	0	5	15	-5	0	0

При решении задачи возможны многие варианты программ. Рассмотрим на примере упрощенной схемы бруса (рис. 2) алгоритм одного из них.

ПОСТАНОВКА ЗАДАЧИ

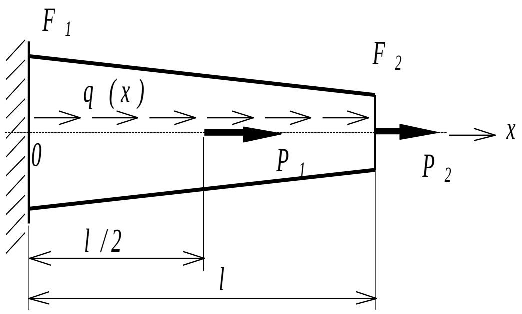


Рис. 2

Исходные данные: $l = 1$м , $F_1 = 4 \cdot 10^{-4}$ м ^{2}, $F_2 = 2 \cdot 10^{-4}$ м ^{2}, $P_1 = 10$ Кн , $P_2 = -20$ Кн , $k = 5$ Кн / м ^{2}, $b = -5$ Кн / м .

Уравнение изменения площади поперечного сечения бруса F(x) запишется в виде

$$F(x) = -2 \cdot 10^{-4} x + 4 \cdot 10^{-4}$$

Закон изменения распределенной нагрузки

$$q(x) = 5x - 5$$

Так как при $x = l / 2$ в точке приложения силы P_1 продольная сила N терпит разрыв, брус необходимо разбить на два силовых участка: 1-ый участок $0 < x < l / 2$; 2-ой участок $l / 2 < x < l$. При этом в точке $x = l / 2$ величины N и σ необходимо вычислить дважды: слева и справа от точки.

АЛГОРИТМ РЕШЕНИЯ.

Введем в рассмотрение массив $X(6)$, элементы которого числено равны координатам точек в которых вычисляются N , σ , u . При трех точках на каждом участке и равномерной сетке получим:

$$x_1 = 0, \quad x_2 = l / 4, \quad x_3 = l / 2, \quad x_4 = l / 2, \quad x_5 = 3l / 4, \quad x_6 = l .$$

Для вычислений и хранения усилий, напряжений и перемещений в точках x_i , будем использовать массивы $N(6)$, $\sigma(6)$, $u(6)$.

Вычисление усилий методом сечений удобно начинать со свободного конца бруса. Из расчетной схемы видно, что

$$N_6 = P_2 .$$

Для N_5 из (1) имеем:

$$N_5 = N_6 + \int_{x_5}^{x_6} q(x) dx$$

Вычисляя последний интеграл по формулам трапеций с использованием значений функции $q(x)$ только на концах отрезка интегрирования (т. к. $q(x)$ - линейная функция, то результат будет точным), получим

$$N_5 = N_6 + [q(x_6) + q(x_5)]\, h/2$$

где $h = x_6 - x_5$ - шаг интегрирования. В рассматриваемом случае для равномерности сетки независимо от пределов интегрирования шаг постоянен и равен $h = l/4$.

Для N_4 аналогично получим формулу

$$N_4 = N_5 + [q(x_5) + q(x_4)]\, l/8$$

В точке разрыва имеем: $N_3 = N_4 + P_1$

и далее: $N_2 = N_3 + [q(x_3) + q(x_2)]\, l/8$

$$N_1 = N_2 + [q(x_2) + q(x_1)]\, l/8$$

Если увеличить длину массива N на единицу и ввести значение $N_7 = 0$, то вместо $N_6 = P_2$ получим $N_6 = N_7 + P_2$.

Тогда можно независимо от количества силовых участков и точек разбиения внутри участков записать следующие рекуррентные формулы для определения усилий в поперечных сечениях бруса:

$N_{m\cdot k+1} = 0$ - значение усилий за пределом бруса;

$N_{j\cdot k} = N_{j\cdot k+1} + P_j$, j = m, m-1, ... ,1 - при переходе через j-ый стык (стык j-го и j+1 -го силовых участков);

$N_i = N_{i+1} + [q(x_{i+1}) + q(x_i)]\, h/2$, $i = j\cdot k + 1, \dots, (j-1)\cdot k + 1$ - для точек внутри j-го силового участка.

Здесь: m - число участков; k - количество точек на участке; h - шаг разбиения. Если при разбиении используется неравномерная сетка, то $h_i = x_{i+1} - x_i$.

Знач усилия и напряжения (последние определяются в соответствии с формулой (2)), можно вычислить перемещения в контрольных точках по формуле (3). При интегрировании по формуле трапеций (в данном случае это приводит к определенной ошибке) получим:

$u_1 = 0$, $u_2 = u_1 + [\sigma_1 + \sigma_2]\, l/(8E)$, $u_3 = u_2 + [\sigma_2 + \sigma_3]\, l/(8E)$

$u_4 = u_3$, $u_5 = u_4 + [\sigma_4 + \sigma_5]\, l/(8E)$, $u_6 = u_5 + [\sigma_5 + \sigma_6]\, l/(8E)$

или в общем случае:

$u_1 = 0$, - в начале отрезка;

$u_i = u_{i-1} + [\sigma_{i-1} + \sigma_i]\, h/(2E)$, i от $(j-1)\cdot k + 2$ до $j\cdot k$ с шагом 1- внутри j-го участка; (6)

$u_{j\cdot k+1} = u_{j\cdot k}$, j = m, m-1, ... ,1 - при переходе через j-й стык . (7)

Для обобщения формулы (7) на последний участок необходимо увеличить длину массива u до m+1.

После того, как выписаны все необходимые формулы, можно предложить следующий алгоритм вычислений:

- Описание массивов X (6) , N (7) , σ (6) , u (7) , P (6) .
- Описание функций $F(x)$, $q(x)$.
- Задание исходных данных (массивов X , P, значений k , m , l).
- Вычисление элементов массива N :σ

внешний цикл по j от m до 1 с шагом -1

$N_{j\cdot k} = N_{j\cdot k+1} + P_j$

внутренний цикл по i от $j\cdot k + 1$ до $(j-1)\cdot k + 1$ с шагом -1

$N_i = N_{i+1} + [q(x_{i+1}) + q(x_i)]\, h / 2$

конец внутреннего цикла

конец внешнего цикла.

- Вычисление элементов массива σ по формулам (2) :

  цикл по i от 1 до m·k с шагом 1

- Вычисление элементов массива u :

  $u_1 = 0$

  внешний цикл по j от 1 до m с шагом 1

  внутренний цикл по i от (j -1)·k +2 до j·k с шагом 1

  $u_i = u_{i-1} + [\sigma_{i-1} + \sigma_i]\, h / (2E)$

  конец внутреннего цикла

  $u_{j \cdot k+1} = u_{j \cdot k}$

  конец внешнего цикла.

- Печать значений x_i, N_i, σ_i, u_i :

  цикл по i от 1 до 6 с шагом 1

В качестве альтернативного можно предложить следующий вариант решения задачи. Интеграл, входящий в формулу для усилий (1) , вычислить аналитически. Это позволяет выражение для вычислений напряжений оформить в программе как функцию. Перемещения при этом можно вычислять не по формуле трапеций по трем точкам на участке (как в предыдущей программе), а используя более точный метод (например, стандартную программу метода Гаусса QG8).

ОПРЕДЕЛЕНИЕ СОБСТВЕННЫХ ЧАСТОТ КОЛЕБАНИЙ СИСТЕМЫ С НЕСКОЛЬКИМИ СТЕПЕНЯМИ СВОБОДЫ

ЗАДАНИЕ для пружинно-массовой механической системы методом простых итераций вычислить высшую и низшую частоты и соответствующие формы колебаний системы.

ПОСТАНОВКА ЗАДАЧИ Пусть имеется механическая система, состоящая из n масс m_j ($j = 1, 2, \dots ,n$), соединенных между собой и основанием пружинами жесткостью c_k . На рисунке 1 показана одна из таких систем для $n = 3$. Массы могут двигаться только вертикально, без поворотов. В этом случае число степеней свободы и, соответственно, число обобщенных координат, необходимых для описания положения системы равно n . Выберем в качестве обобщенных координат смещения x_j ($j = 1, 2, \dots ,n$) сосредоточенных масс m_j от положения равновесия.

Уравнения свободных колебаний системы около положения равновесия можно

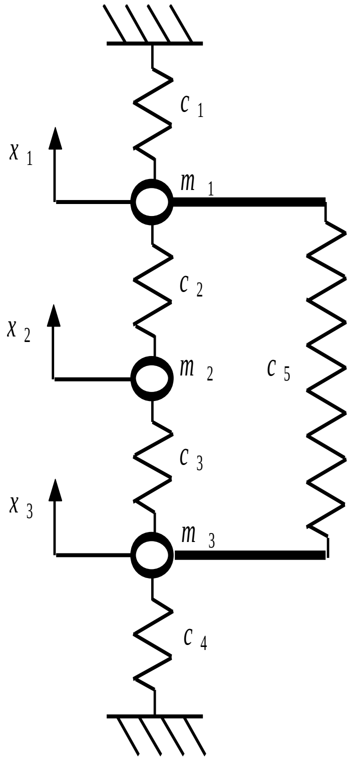


Рис. 1.

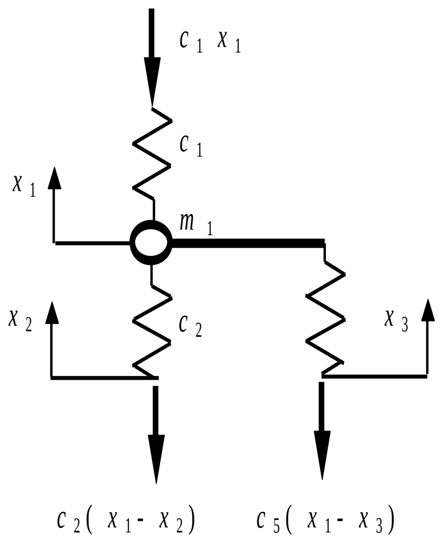


Рис. 2.

получить, используя II-й закон Ньютона.

Запишем уравнения II-й закона Ньютона для каждой массы

$$m_j \ddot{x}_j = F_j, \qquad (j = 1,2,\dots,n), \qquad \ddot{x}_j = \frac{d^2 x_j}{d\ t^2}.$$

Восстанавливающие силы F_j , вычисляются как суммы сил, действующие на j-ю массу со стороны пружин, соединенных с ней, при отклонении массы от положения равновесия,. Усилие в каждой пружине численно равно произведению жесткости пружины (c_j) на ее удлинение (разности смещений концов пружины).

Так, на массу m_1 системы (рис. 1) действуют три силы (рис. 2): со стороны верхней пружины - $c_i\, x_1$ (знак «минус» показывает, что сила действует в противоположную сторону от направления x_1), со стороны левой нижней пружины - $c_2\, (x_1 - x_2)$ (здесь $(x_1 - x_2)$ - удлинение пружины), со стороны правой нижней пружины - c_5 (x_1 - x_3). Тогда уравнение движения для m_1 имеет вид

$$m_1\ddot{x}_1 = -c_1x_1 - c_2(x_1 - x_2) - c_5(x_1 - x_3)$$

Аналогично можно получить уравнения движения для всех масс. В итоге имеем систему уравнений движения

$$\begin{aligned} m_1\ddot{x}_1 &= -c_1x_1 - c_2(x_1 - x_2) - c_5(x_1 - x_3) \\ m_2\ddot{x}_2 &= -c_2(x_2 - x_1) - c_3(x_2 - x_3) \\ m_3\ddot{x}_3 &= -c_4x_3 - c_3(x_3 - x_2) - c_5(x_3 - x_1) \end{aligned} \qquad (1)$$

Решение системы дифференциальных уравнений (1) разыскивается в виде

$x_k = X_k e^{i\omega t}$, $(k = 1, 2, 3)$, (2)

где: X_k -амплитуда колебаний k-ой массы, ω - круговая частота.

Подставляя функции (2) в уравнения (1) , и сокращая на $e^{i\omega t}$, получим систему линейных однородных алгебраических уравнений

$$\begin{aligned} m_1\omega^2 X_1 &= c_1X_1 + c_2(X_1 - X_2) + c_5(X_1 - X_3) \\ m_2\omega^2 X_2 &= c_2(X_2 - X_1) + c_3(X_2 - X_3) \\ m_3\omega^2 X_3 &= c_4X_3 + c_3(X_3 - X_2) + c_5(X_3 - X_1) \end{aligned} \qquad (2)$$

Вводя обозначения $\lambda = c_1 / m_1\omega^2$, представим систему (1) следующим образом

$$X_1 = \lambda\,[(1 + \frac{c_2}{c_1} + \frac{c_5}{c_1})X_1 - \frac{c_2}{c_1}X_2 - \frac{c_5}{c_1}X_3]$$

$$X_2 = \lambda\,[-\frac{c_2}{c_1}\frac{m_1}{m_2}X_1 + (\frac{c_2}{c_1} + \frac{c_3}{c_1})\frac{m_1}{m_2}X_2 - \frac{c_3}{c_1}\frac{m_1}{m_2}X_3]$$

$$X_3 = \lambda\,[-\frac{c_5}{c_1}\frac{m_1}{m_3}X_1 - \frac{c_3}{c_1}\frac{m_1}{m_3}X_2 + (\frac{c_3}{c_1} + \frac{c_4}{c_1} + \frac{c_5}{c_1})\frac{m_1}{m_3}X_3]$$

или в матричном виде

$$\{X\} = \lambda\,[H]\{X\} \qquad (3)$$

где $\{X\}^{T} = \{X_1, X_2, X_3\}$, а элементы матрицы $[H]$ вычисляются по формулам:

$$H_{11} = 1 + \frac{c_2}{c_1} + \frac{c_5}{c_1}, \quad H_{12} = -\frac{c_2}{c_1}, \quad H_{13} = -\frac{c_5}{c_1},$$

$$H_{21} = -\frac{c_2}{c_1}\frac{m_1}{m_2}, \quad H_{22} = (\frac{c_2}{c_1} + \frac{c_3}{c_1})\frac{m_1}{m_2}, \quad \text{и т.д.}$$

Для определения минимального собственного значения λ и соответствующего ему собственного вектора $\{X\}$ системы (3) можно воспользоваться несколькими методами, в частности, методом простых итераций, который заключается в следующем.

Выбирается некоторое начальное приближение для собственного вектора

$\{X\}^{(0)\,T} = \{X_1^{(0)}, X_2^{(0)}, X_3^{(0)}\}$

по формуле

$$\{X\}^{(1)} = [H]\{X\}^{(0)}$$

или в общем случае

$$\{X\}^{(i+1)} = [H]\{X\}^{(i)}$$

На каждом шаге итераций определяются $\lambda_k^{(i+1)} = X_k^{(i)} / X_k^{(i+1)}$, $k = 1, 2, \dots, n$, С ростом числа итераций $\lambda_k^{(i+1}$ сходятся к минимальному собственному значению λ системы (3) . При достижении определенной точности в вычислении λ процесс итерации прекращается.

В процессе итераций абсолютные значения компонент вектора{ X } растут, поэтому при численных расчетах после каждого шага итераций необходимо выполнять нормирование{ X }, например, по значению максимального модуля его элементов.

ЗАДАНИЕ. Для своего варианта

1. Получить систему дифференциальных уравнений движения в виде (1).
2. Записать систему однородных алгебраических уравнений в виде (2), (3). Выписать выражения для определения элементов матрицы [H].
3. Составить подпрограмму, в которой методом простых итераций вычисляются минимальные значения λ и соответствующий собственный вектор{ X } для системы (3).
4. Составить головную программу для вычислений min λ (для матрицы [H]) и max λ (для матрицы [H]$^{-1}$).

АЛГОРИТМ РЕШЕНИЯ

1. Составить подпрограмму. в которой вычисляются компоненты матрицы [H].
2. Составить головную программу, в которой:
   - вводятся исходные данные ($\frac{c_2}{c_1}$, $\frac{c_3}{c_1}$, … , $\frac{m_1}{m_2}$, …)
   - вызывается подпрограмма вычисления [H]
   - вызывается подпрограмма метода простых итераций
   - выводятся результаты (λ , { X }, число итераций)

Данные из головной программы в подпрограмму вычисления [H] можно передавать либо через список формальных параметров, либо через общие области, используя оператор COMMON.

Подпрограмму метода простых итераций можно организовать следующим образом.

Формальные параметры H, X, Y, Z, AL, N, E ,m, mmax, IER где:

входные параметры: H (N, N) - матрица системы (3), N - порядок системы, E - точность вычислений; mmax - максимальное число итераций;

выходные параметры: AL - значение $\lambda_{\min}$ системы (3), X(N) - соответствующий $\lambda_{\min}$ собственный вектор; m - число выполненных итераций; IER - код ошибки (IER = 0 - процесс сошелся, в противном случае IER=1).

массивы: Y(N), Z(N) - вспомогательные массивы для хранения результатов промежуточных вычислений.

Алгоритм подпрограммы

a) описание используемых массивов
b) задание начальных значений X
c) вычисление последующих приближений Y=H·X
d) вычисление значений $Z_i = X_i / Y_i$, $i = 1,\dots, n$
e) найти среднеквадратичную ошибку

$$E1 = \left| \frac{1}{|z_n|} \frac{1}{n-1} [\sum_{i=1}^{n-1} (z_i - z_n)^2]^{1/2} \right|$$

f) найти максимальный модуль элемента Y
   $S = \max(|Y_1|, |Y_2|, \dots, |Y_n|)$

g) вычислить начальное приближение для следующего шага итераций
   $X_i = Y_i / S$, $i = 1, n$
h) выполнить проверку на прекращение итераций:
   если $E1 - E > 0$, перейти на пункт c)
i) выполнить проверку на прекращение итераций по mmax:
   если m= mmax , выйти из подпрограммы с кодом ошибки IER=1
j) запомнить вычисленное значение λ (последнее значение Z_1)
   $AL = Z_1$
k) вернуться в вызывающую программу.

ВАРИАНТЫ ЗАДАНИЯ. Исходная расчетная схема представляет собой цепочку из 5-ти масс и 6-ти жесткостей. В зависимости от варианта, одна из пружин может иметь нулевую жесткость (не учитывается при выводе уравнений движения),а некоторые массы могут быть соединены между собой или с основанием дополнительной "параллельной" жесткостью. В таблице 1 приведены номер нулевой жесткости и точки, соединенные "параллельной" жесткостью. В таблице 2 в зависимости от последней цифры кода группы заданны значения отношений масс и нулевых жесткостей.

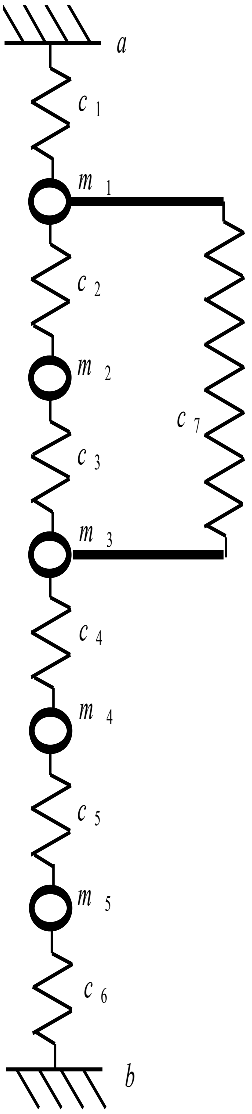


Рис. 3.

Таблица 1. Варианты расчетных схем.

№ вар.	Нулевые жесткости	Парал-ые жесткости	№ вар.	Нулевые жесткости	Парал-ые жесткости
1	2	4	1	2	4
1	1	—	16	—	4-b
2	1	1-3	17	—	a-3
3	1	2-4	18	—	1-4
4	1	3-5	19	—	2-5
5	1	4-b	20	—	1-5
6	1	1-4	21	2	1-3
7	1	2-5	22	3	2-4
8	1	3-b	23	4	3-5
9	1	1-5	24	3	1-4
10	1	2-b	25	3	1-5
11	—	—	26	1	2-4
12	—	1-3	27	1	1-3
13	—	2-4	28	1	2-4
14	—	3-5	29	1	2-5
15	—	a-2	30	1	2-5

Таблица 2. Варианты жесткостей и масс.

№	Жесткости c_i / c i						Массы m_i / m				
вар.	i=1	i=2	i=3	i=4	i=5	i=6	i=1	i=2	i=3	i=4	i=5
1	1.0	2.0	0.8	1.2	0.9	1.5	1.0	0.5	1.2	2.0	0.8
2	1.0	1.5	0.5	2.0	2.0	1.0	1.0	1.0	2.0	3.0	0.5

ОПРЕДЕЛЕНИЕ СОБСТВЕННЫХ ФОРМ КОЛЕБАНИЙ УПРУГОЙ БАЛКИ.

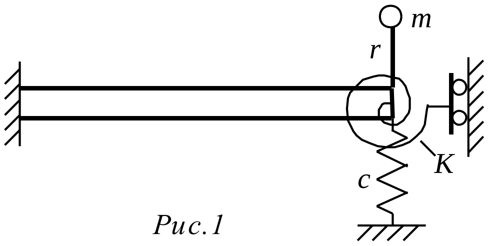


Рис.1

Обозначения:

$Y(x)$ - прогиб оси балки;

EJ - жесткость балки на изгиб;

l - длина балки;

ρF - погонная масса;

m - сосредоточенная масса;

r - длина жесткого рычага - расстояние между осью балки и массой m;

c - линейная жесткость упругой опоры;

K - угловая жесткость упругой опоры;

$z = x/l$ - безразмерная координата ($0 \le z \le 1$).

При решении задач на определение собственных форм колебаний упругих балок функцию прогиба удобно записывать через функции Крылова $\mathrm{S}(x)$, $\mathrm{T}(x)$, $\mathrm{U}(x)$, $\mathrm{V}(x)$, которые представляют собой линейные комбинации гиперболических и тригонометрических синусов и косинусов:

$$\begin{aligned} &\mathrm{S}(x) = [\mathrm{ch}(x) + \cos(x)]\,/\,2, \quad \mathrm{T}(x) = [\mathrm{sh}(x) + \sin(x)]\,/\,2 \\ &\mathrm{U}(x) = [\mathrm{ch}(x) - \cos(x)]\,/\,2, \quad \mathrm{V}(x) = [\mathrm{sh}(x) - \sin(x)]\,/\,2. \end{aligned} \tag{1}$$

Собственные формы $Y(z)$ колебаний консольной балки (рис.1) через функции Крылова определяются как

$$Y(z) = \mathrm{U}(\alpha z) + D\mathrm{V}(\alpha z) \tag{2}$$

Параметры α и D зависят от граничных условий на правом конце балки. Собственные значения α (безразмерные собственные частоты) определяются из следующего характеристического уравнения:

$$\begin{vmatrix} \mathrm{V}(\alpha) + \left(f\alpha - \dfrac{g}{\alpha^3}\right)\mathrm{U}(\alpha) & \mathrm{S}(\alpha) + \left(f\alpha - \dfrac{g}{\alpha^3}\right)\mathrm{V}(\alpha) \\ \mathrm{S}(\alpha) - \left(w\alpha^3 - \dfrac{q}{\alpha}\right)\mathrm{T}(\alpha) & \mathrm{T}(\alpha) - \left(w\alpha^3 - \dfrac{q}{\alpha}\right)\mathrm{U}(\alpha) \end{vmatrix} = 0 \tag{3}$$

где f,g,w,q-безразмерные параметры: $f = m/\rho Fl$, $g = cl^3/EJ$, $w = f(r/l)^2$, $q = Kl/EJ$.

Уравнение (3) имеет бесчисленное множество положительных корней α , i =1,2,3, ... Для каждого значения а коэффициент D определяется из уравнения

$$\mathrm{V}(\alpha) + \left(f\alpha - \frac{g}{\alpha^3}\right)\mathrm{U}(\alpha) + D\left[\mathrm{S}(\alpha) + \left(f\alpha - \frac{g}{\alpha^3}\right)\mathrm{V}(\alpha)\right] = 0 \tag{4}$$

Каждой паре значений α и D соответствует своя собственная форма Y(z), которая определяется с точностью до множителя. Собственные формы можно пронормировать, например, так, чтобы максимальное значение нормированной формы было равно единице. Тогда для нормированной формы можно записать следующее выражения

$$Y(z) = Y(z)/B = [\mathrm{U}(\alpha z) + D\mathrm{V}(\alpha z)] \,/\, B \qquad (5)$$

$$B = \max|Y(z)| \ (6)$$

Точкой α , в которой функция $Y(z)$ принимает максимальное по модулю значение B, может быть либо экстремум функции, либо точка z =1. В первом случае для определения z необходимо найти корни уравнения $Y'(z)$=0, или в развернутом виде

$$\mathrm{T}(\alpha z) + D\mathrm{U}(\alpha z) = 0 \qquad (7)$$

Так как корней на отрезке 0 < z < 1 может быть несколько (в вариантах задания - не более трех), то необходимо найти все. Затем из найденных точек и точки z = 1 выбрать в качестве z ту, в которой $|Y(z)|$ принимает максимальное значение.

Можно использовать и другой способ нормирования, при котором коэффициент В вычисляется по формуле

$$B = \int_0^1 [Y(z)]^2 \, dz \qquad (8)$$

Если собственная форма имеет узлы (точки, где функция $Y(z)$ меняет знак), то интеграл от 0 до 1 надо вычислять как сумму интегралов по участкам между узлами. Поэтому сначала необходимо найти эти узлы, т.е. корни уравнения (2). В зависимости от варианта задания на отрезке от 0 до 1 их может быть не более двух.

ЗАДАНИЕ. Для своего варианта исходных данных вычислить три первых собственных значения α из уравнения (3) и построить графики соответствующих собственных нормированных форм.

ВЫБОР ВАРИАНТА ЗАДАНИЯ. Переведите номер по групповому журналу из десятичной в двоичную систему исчисления. Запишите перед полученным числом нули так, чтобы получился пятизначный код. По двоичным значениям в каждой позиции кода определяются исходные данные (до 23-го варианта):

N позиции (слева направо)	1	2	3	4	5
Параметр	f	w	g	q	способ нормир.
0 в позиции	1	0	0	0	(6)
1 в позиции	0	1	1	1	(8)

РЕКОМЕНДАЦИИ к составлению программы. Для решения нелинейных уравнений можно использовать подпрограммы отделения корней и метода половинного деления (см. задачу 5), для численного интегрирования - подпрограмму метода Гаусса (см. задачу 6). Необходимо также составить подпрограммы-функции и (или) подпрограммы для вычисления: функций Крылова; функций (2),(3),(5),(7); подынтегральной функции в формуле (8); коэффициента B по формулам (6) либо (8).

В головной программе требуется описать массивы:

- A(3) -= для собственных значений α;
- D(3) - для коэффициентов *D*;
- Z0(3) - для точек экстремумов (или узлов);
- Y0(3) – для значений функции *Y* в точках экстремумов;
- B(3) - для коэффициентов *B* в формулах (6) или (8).

Алгоритм может быть построен по следующей схеме. В цикле по номеру формы (i=1,2,3) вычисляются:

- A(i) - собственное значение α;
- D(i) - коэффициент *D*;
- k значений Z0 и Y0 - точки экстремумов i-ой формы и значения функции *Y* в этих точках ($k \le 3$ - зависит от номера формы) или k значений координат узлов i-ой формы ($k \le 2$);
- B(i) - коэффициент *B* по формуле (6) или (8). Все результаты выводятся на печать.

После завершения цикла по i вычисляются при изменении *z* от 0 до 1 с шагом не более 0.1 и печатаются в таблицу ординаты трех собственных форм. При этом для записи и хранения текущих значений можно использовать массив Z0 или Y0.

ПРИМЕНЕНИЕ КОМПЛЕКСНЫХ ВЕЛИЧИН.

ВВЕДЕНИЕ. При выполнении расчетов с использованием методов ТФКП, теории упругости и т.д. в ФОРТРАН-программах могут быть использованы комплексные величины. Объектом комплексного типа в ФОРТРАНе могут быть константа, переменная, элемент массива, результат обращения к функции, значение выражения.

Данное комплексного типа представляется парой значений вещественного типа. Первая компонента пары представляет действительную часть, а вторая - мнимую часть комплексного данного. Комплексные константы определяются по форме записи - это пара вещественных констант, разделенных запятой и заключенных в скобки; первая вещественная константа задает действительную часть комплексного числа, вторая - мнимую.

Комплексные переменные, массивы, внешние функции должны быть описаны в операторе описания типа COMPLEX. Например, оператор COMPLEX ZZ, Z1(4,4) описывает комплексную величину ZZ (переменную либо функцию) и двумерный массив Z1 , состоящий из ·16 комплексных элементов.

В ФОРТРАНе имеются встроенные функции, предназначенные для работы с комплексными величинами (ниже в таблице Z - комплексная, а X и Y - вещественные величины):

Тип функции	Функция	Значение
комплексный	CMPLX(X,Y)	X+i·Y
----’’----	CONJG(Z)	X-i·Y
----’’----	CSIN(Z)	sin(Z)
----’’----	CCOS(Z)	cos(Z)
----’’----	CEXP(Z)	exp(Z)
----’’----	CSQRT(Z)	sqrt(|Z|)·exp(i·arg(Z)/2)
----’’----	CLOG(Z)	ln(|Z|) +i·arg(Z)
вещественный	REAL(Z)	Re(Z)
----’’----	AIMAG(Z)	Im(Z)
----’’----	CABS(Z)	$\|Z\|=\mathrm{sqrt}(X^2+Y^2)$
----’’----	ATAN2(Y,X)	arg(Z)=arctg(Y/X)

CSQRT(Z) sqrt(|Z|)·exp(i·arg(Z)/2) $-\pi \le \arg(Z) \le \pi$

CLOG(Z) вычисляет главное значение (k=0) функции $\ln(Z) = \ln(|Z|)+i\cdot(\arg(Z)+2\cdot k\cdot\pi)$

Для вычисления других комплексных функций можно воспользоваться формулами:

$tg(Z) = \sin(Z)/\cos(Z)$ $\quad$ $arctg(Z) = (1/(2\cdot i))\cdot\ln((1+i\cdot Z)/(1-i\cdot Z))$

$ctg(Z) = \cos(Z)/\sin(Z)$ $\quad$ $arcctg(Z) = \pi/2-(1/(2\cdot i))\cdot\ln((1+i\cdot Z)/(1-i\cdot Z))$

$th(Z) = sh(Z)/ch(Z)$ $\quad$ $arcth(Z) = 0.5\cdot\ln((1+Z)/(1-Z))$

$cth(Z) = ch(Z)/sh(Z)$ $\quad$ $arccth(Z) = 0.5\cdot\ln((Z+1)/(Z-1))$

$sh(Z) = 0.5\cdot(\exp(Z)-\exp(-Z))$ $\quad$ $arcsh(Z) = \ln(Z \pm (Z^2+1)^{0.5})$

$ch(Z) = 0.5\cdot(\exp(Z)+\exp(-Z))$ $\quad$ $arcch(Z) = \ln(Z \pm (Z^2-1)^{0.5})$

$\arcsin(Z) = \pi/2-i\cdot\ln(Z+(Z^2-1)^{0.5})$ $\quad$ $\arccos(Z) = i\cdot\ln(Z \pm (Z^2-1)^{0.5})$

Для функций, при вычислении которых используется CLOG(Z), определяются их главные значения.

В арифметическом выражении совместно с комплексными операндами могут быть использованы величины любого типа. При вычислении выражений со смешением типов применяются правила преобразования типов. Так, при выполнении арифметической операции между операндами, один из которых является комплексным, результат будет также комплексным. Запрещены арифметические операции между комплексными величинами и вещественными удвоенной точности. При выполнении оператора присваивания тип вычисленного значения в правой части оператора преобразуется к типу переменной в левой части.

При вводе-выводе комплексных данных необходимо задавать формат как для действительной части, так и для мнимой.

ЗАДАНИЕ 1. ВЫЧИСЛЕНИЕ КОМПЛЕКСНЫХ ФУНКЦИЙ. Для восьми значений комплексной переменной Z вычислить комплексные функции:

1) Z, |Z|, agr(Z), Re(Z), Im(Z), sin(Z), cos(Z), exp(Z), sqrt(Z), ln(Z) - для всех вариантов;

2) f1, f2, f3, f4 в соответствии с номером варианта из таблицы:

n	f1	f2	f3	f4
1	$Z^{2 \cdot i}$	tg(Z) / th(1+i)	arctg((2- 0.5·i)·Z)	cos(-2·i) · ch(Z)
2	$Z^{1/3}$	ctg(i) / cth(Z)	arcctg((0.1+ i)·Z)	sin(0.1·i) · sh(Z)
3	$(2+i)^{Z}$	th(Z) / tg(3- i)	arcsin((-1- i)·Z)	(2-i) · sh(i·Z)
4	$(i \cdot Z)^{1/4}$	(3-0.4·i)· th(Z)	arccos(0.1-0.3·i·Z)	ch((0.2-i)-2·Z)
5	$(1+Z)^{2 \cdot i}$	cth(0.5·i · (Z+i))	arcth(Z / (2- i))	sh((1- i)·(Z+1))
6	$(2 \cdot i+Z)^{1/Z}$	th(Z) / th(-i)	arccth(i / Z)	ch(Z+i) / sin(i)
7	$Z^{i/2}$	ctg(Z / (2+0.1·i))	arcsh(i + i / Z)	sh(i -Z)·cos(i)
8	$(1+Z)^{1/2}$	tg(Z / (1+Z))	arcch(Z / (1+ i))	sh(Z)·ch(1+Z)
9	$(i+Z)^{1/2}$	th((2-i) · Z)	arctg((2- i)·Z)	sh(Z/(1-i))
10	Z^{1+i}	th(Z / (3+ i))	arcctg(Z·(1-2·i))	ch(Z·(Z+1))
11	$(2 \cdot i+Z)^{0.5}$	th(1+i) / tg(Z)	arcsin((Z+ i) / Z)	cos(i)·ch(Z)
12	$(1-i)^{Z+1}$	cth(Z) / ctg(i)	arccos(Z / (Z - i))	sh(i)·cos(Z)
13	2^{1+i+Z}	th(i-Z) / ctg(i)	arcth(Z^2 +1)	sh(i)·sin(Z)
14	$(Z+1)^{Z-i}$	cth(2·i) / tg(i·Z)	arccth(Z+Z / i)	cos(Z+1)/ch(Z)
15	$(-1+i)^{Z}$	ctg(1- i)·th(Z)	arcsh(Z / (Z+2))	sin(2-i)·ch(Z)
16	Z^{-1-i}	th(Z+1) / th(Z+ i)	arcch((1- i) / Z)	cos(2-i)·ch(Z)

Значения комплексной переменной Z_k вычислить в соответствии с номером n варианта:
$Z_1 = n$; $Z_2 = 1+ i \cdot n$; $Z_3 = i \cdot n$; Z 4= - n+i; $Z_5 = - n$; $Z_6 = - n /3-i \cdot 2 \cdot n /6$; $Z_7 = -i \cdot n$; $Z_8 = 2 \cdot n /3-i \cdot 4 \cdot n /5$

ЗАДАНИЕ 2. КОНФОРМНОЕ ОТОБРАЖЕНИЕ. Для области D в плоскости Z построить отображенную функцией W= f(Z) область D* в плоскости W. Для этого:

1. записать уравнения, описывающие границы области D;
2. выбрать (вычислить) на границах области D не менее 20 точек Zк таким образом, чтобы при их обходе область оставалась всегда слева;
3. вычислить точки Wк = f(Zк) области D*;
4. по найденным точкам на миллиметровке построить области D и D*.

Варианты заданий.

N	D	W
1	$\lvert Z \rvert < 1$	(Z+2)/(Z+i)
2	$-\pi/2 < Re(Z) < \pi/2$	tg(Z)
3	$\lvert Z \rvert > 3$	(Z+1/Z)/2
4	$-\pi/4 < Im(Z) < \pi/4$	th(Z)

5	$\|Z\| < 0.5$	$(Z+1/Z)/2$
6	$0 < \|Z\| < 1$, $0 < \arg(Z) < \pi/3$	$(Z+1/Z)$
7	$\|Z-1\| < 1$	$1/Z$
8	$\|Z\| > 1$, $0 < \arg(Z) < \pi/2$	$i \cdot \ln(Z)$
9	$\|Z-2\| > 2$	$1/(Z-1)$
10	$0 < \mathrm{Im}(Z) < 2\pi$, $\mathrm{Re}(Z) < 0$	$i \cdot \exp(Z)$
11	$0 < \mathrm{Im}(Z) < \pi$	$\mathrm{ch}(Z)$
12	$0 < \mathrm{Re}(Z) < \pi/2$, $0 < \mathrm{Im}(Z) < 1$	$\exp(2i \cdot Z)$
13	$0 < \mathrm{Re}(Z) < \pi/4$	$\mathrm{ctg}(Z)$
14	$-\pi < \mathrm{Re}(Z) < \pi$, $\mathrm{Im}(Z) > 0$	$\sin(Z)$
15	$-\pi < \mathrm{Im}(Z) < \pi$	$Z+\exp(Z)$
16	$\mathrm{Re}(Z) > 0$, $-1 < \mathrm{Im}(Z) < 0$	$\mathrm{ch}(Z)$

Лабораторные работы 4-го семестра соответствуют задачам курсовой работы по сопротивлению материалов. Студенты составляют программы, выполняют расчеты и строят эпюры с использованием программных продуктов **Microsoft Excel, MathCAD, Fortran**. Отчет оформляется в Word. Варианты заданий приведены во второй части приложения.

Пример **1. РАСЧЕТ ПРОГИБА ШАРНИРНО-ОПЕРТОЙ БАЛКИ**

1.1 Постановка задачи

Задана балка, закрепленная на двух шарнирных опорах, и нагружена в соответствии с расчетной схемой (рис. 1.1). Для данной балки построить эпюры прогибов, углов поворота, моментов и перерезывающих сил, используя три программы Fortran, Mathcad, Excel. Исходные данные приведены в таблице 1.1.

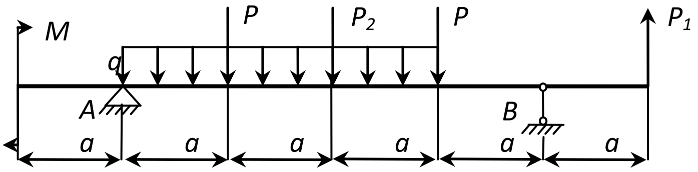


Рис. 1.1 Шарнирно-опертая балка

Исходные данные. Таблица 1.1.

Параметры	*Значение параметров*
M_1, кНм	40
P_1, кН	-10
P_2, кН	-20
P_3, кН	10
q, кН/м	-10
a, м	0,5
EI	1

1.2 Разработка алгоритма вычисления функции прогиба

Изменим расчетную схему на рис. 1.1. На новой схеме (рис. 1.2) направим все силы и все реакции вверх. Распределенную нагрузку продолжим до конца, т.к. она должна заканчиваться на правом конце балки, и уравновесим ее другой распределенной нагрузкой, направленной в противоположную сторону.

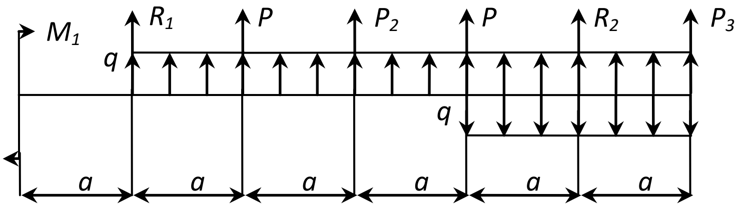


Рис. 1.2 Расчетная схема

Выберем систему координат так, чтобы начало ее совпадало с левым концом балки (рис. 1.2), разобьем балку на силовые участки.

Текст программы в Mathcad:

Для начала введем все исходные данные из таблицы 1.1. В Mathcad'е существует операция присваивания, которая позволяет оценивать любую переменную справа от знака присваивания и назначает любой результат слева от этого знака.

$M1 := 40 \quad P1 := -10 \quad EI := 1 \quad P2 := -20 \quad q := -10 \quad a := 0.5 \quad P3 := 10$

Из условия равновесия найдем реакции опор в точках A и B:

$$R1 := \frac{-M1 - 5q \cdot a \cdot 1.5 \cdot a - P1 \cdot 3 \cdot a - P2 \cdot 2 \cdot a - P1 \cdot a + P3 \cdot a}{4a}, \quad R1 = 11.875 ,$$

$$R2 := \frac{M1 - 5q \cdot a \cdot 2.5 \cdot a + 2q \cdot a \cdot 4 \cdot a - P1 \cdot a - P2 \cdot 2a - P1 \cdot 3 \cdot a - P3 \cdot 5a}{4a}, \quad R2 = 33.125 .$$

Далее вводим функцию z, которая должна удовлетворять условию: $z(x,a) = \begin{cases} 0, & x < a \\ 1, & x \ge a \end{cases}$. Это необходимое условие, для того чтобы универсальное уравнение было верно для каждого участка.

В Mathcad это будет иметь вид:

$z(x,a) := \text{if}(x < a, 0, 1).$

Затем вводим функцию *W(x)*, которая будет входить в универсальное уравнение упругой линии, для определения прогибов и углов поворота. Эта функция должна быть больше либо равна нулю, т.к. здесь учитываются силовые факторы, лежащие слева от рассматриваемого сечения.

$$W(x) := M1 \cdot \frac{x^2}{2} + R1 \cdot \frac{(x-a)^3}{6} \cdot z(x,a) + P1 \cdot \frac{(x-2\cdot a)^3}{6} \cdot z(x,2\cdot a) + P2 \cdot \frac{(x-3\cdot a)^3}{6} \cdot z(x,3\cdot a) \ldots$$
$$+ P1 \cdot \frac{(x-4\cdot a)^3}{6} \cdot z(x,4\cdot a) + R2 \cdot \frac{(x-5\cdot a)^3}{6} \cdot z(x,5\cdot a) + P3 \cdot \frac{(x-6\cdot a)^3}{6} \cdot z(x,6\cdot a) \ldots$$
$$+ q \cdot \frac{(x-a)^4}{24} \cdot z(x,a) - q \cdot \frac{(x-4\cdot a)^4}{24} \cdot z(x,4\cdot a).$$

Введем матрицы *D* и *B*:

$$D := \begin{pmatrix} 1 & a \\ 1 & 5a \end{pmatrix}, \quad B := \begin{pmatrix} \frac{-1}{EI} \cdot W(a) \\ \frac{-1}{EI} \cdot W(5a) \end{pmatrix}, \quad B = \begin{pmatrix} -5 \\ -125.026 \end{pmatrix},$$

где **D** – матрица коэффициентов при неизвестных, а **B** – матрица свободных коэффициентов.

Эти матрицы нужны для того, чтобы решить СЛАУ **DC=B**, где **C** – неизвестные коэффициенты.

Для решения СЛАУ используем оператор lsolve

$$C := \text{lsolve}(D,B), \quad C = \begin{pmatrix} 25.007 \\ -60.013 \end{pmatrix}.$$

Задаем функцию, определяющую прогиб:

$$V(x) := C_0 + C_1 \cdot x + \frac{1}{EI} \cdot W(x).$$

Далее строим эпюру прогибов *V(x)*, выбрав на панели инструментов график в декартовых осях. Слева от оси ординат записываем *V(x)*, а на оси абсцисс устанавливаем пределы для координаты x от 0 до 3.

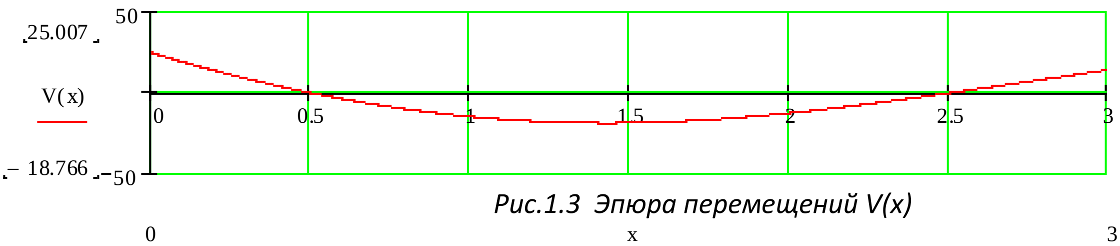


Рис.1.3 Эпюра перемещений V(x)

Продифференцируем в ручную *V(x)* и получим функцию для углов поворота *Q(x)*:

$$Q(x) := C_1 + M1 \cdot x + R1 \cdot \frac{(x-a)^2}{2} \cdot z(x,a) + P1 \cdot \frac{(x-2\cdot a)^2}{2} \cdot z(x,2\cdot a) + P2 \cdot \frac{(x-3\cdot a)^2}{2} \cdot z(x,3\cdot a) \ldots$$
$$+ P1 \cdot \frac{(x-4\cdot a)^2}{2} \cdot z(x,4\cdot a) + R2 \cdot \frac{(x-5\cdot a)^2}{2} \cdot z(x,5\cdot a) + P3 \cdot \frac{(x-6\cdot a)^2}{2} \cdot z(x,6\cdot a) \ldots$$
$$+ q \cdot \frac{(x-a)^3}{6} \cdot z(x,a) - q \cdot \frac{(x-4\cdot a)^3}{6} \cdot z(x,4\cdot a).$$

Точно также как и для прогибов строим эпюру *Q(x)*:

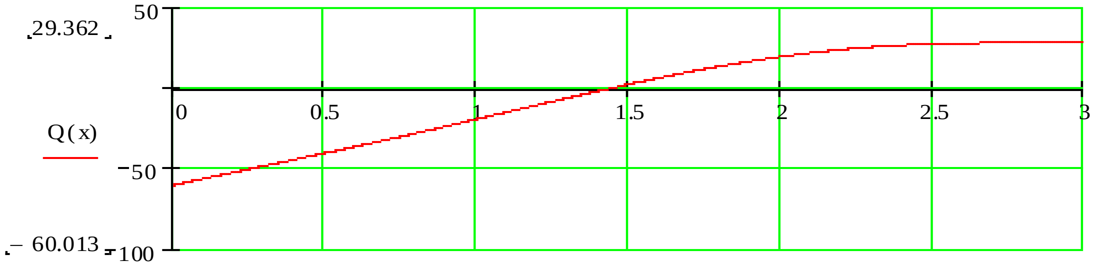


Рис.1.4 Эпюра прогибов балки Q(x)

Далее дифференцируя *Q(x)*, получаем функцию *M(x)*:

$$m(x) := M1 + R1\cdot(x-a)\cdot z(x,a) + P1\cdot(x-2\cdot a)\cdot z(x,2\cdot a) + P2\cdot(x-3\cdot a)\cdot z(x,3\cdot a)\ \ldots$$
$$+ P1\cdot(x-4\cdot a)\cdot z(x,4\cdot a) + R2\cdot(x-5\cdot a)\cdot z(x,5\cdot a) + P3\cdot(x-6\cdot a)\cdot z(x,6\cdot a)\ ..$$
$$+ q\cdot\frac{(x-a)^2}{2}\cdot z(x,a) - q\cdot\frac{(x-4\cdot a)^2}{2}\cdot z(x,4\cdot a)$$

Строим эпюру изгибающих моментов *M(x)*:

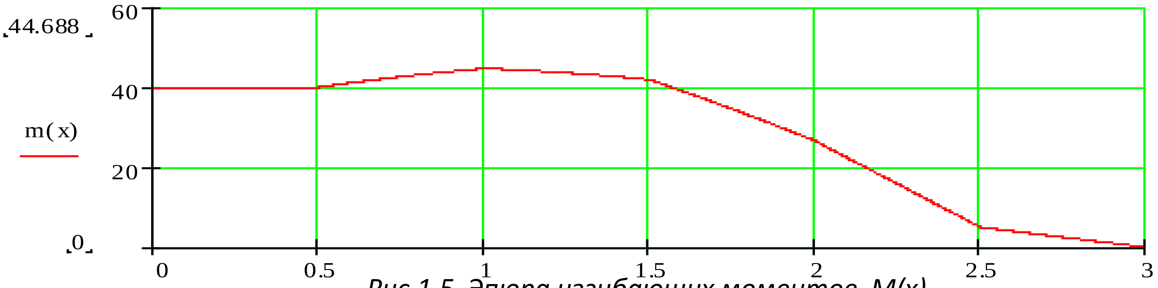


Рис.1.5 Эпюра изгибающих моментов M(x)

Дифференцируя *M(x)*, получаем *P(x)*:

$$P(x) := R1\cdot z(x,a) + P1\cdot z(x,2\cdot a) + P2\cdot z(x,3\cdot a)\ \ldots$$
$$+ P1\cdot z(x,4\cdot a) + R2\cdot z(x,5\cdot a) + P3\cdot z(x,6\cdot a)\ ..$$
$$+ q\cdot(x-a)\cdot z(x,a) - q\cdot(x-4\cdot a)\cdot z(x,4\cdot a)$$

Эпюра для перерезывающих сил *P(x)* имеет вид:

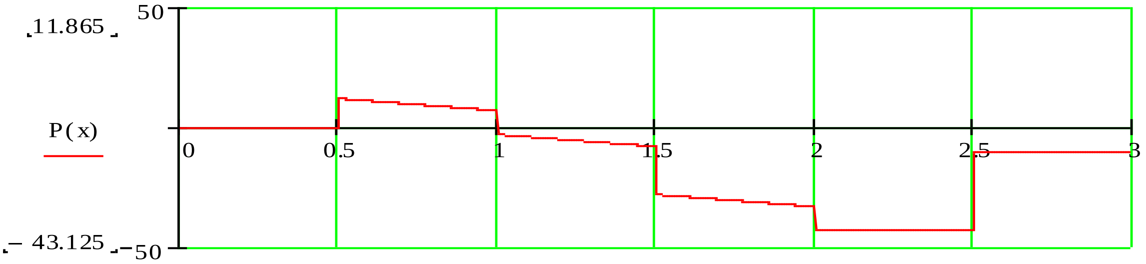


Рис.1.6 Эпюра перерезывающих сил P(x)

Как видно, текст программы очень прост и не требует особых комментариев. Меняя исходные данные, можно пользоваться этой программой повторно. Для построения эпюр других балок придется также изменять уравнения реакции опор.

Алгоритм решения задачи с использованием языка Fortran

1. Описываем заданные массивы
2. Описываем именованные константы, количество силовых участков и количество точек на участке.
3. Задаем функцию *z(i,j)* для каждого участка.
4. Задаем функцию *W(i)*, с использованием исходных данных.
5. Описываем массивы *V(nm)*, *W2(nm)*, *S(nm)*, *F(nm)*, *C*, *D*.

6. Вычисляем координаты узловых точек.
7. Вызываем подпрограмму SIMQ для решения СЛАУ
8. Вычисляем массивы
9. Выводим на экран искомые величины

Вызываем графическую подпрограмму для построения эпюр. Графическая подпрограмма позволяет визуально оценить полученный результат.

Текст программы:

```
module sopromat1
real::P1=-10,P2=-20,q=-10,
P3=10,a=0.5,M1=40,R1,R2, EI=1.
integer, parameter::n=6, m=5, nm=n*m
real(8)::x(nm)
end

Function z(i,j)
z=1
if (j>=i) z=0
end

Function W(i)
use sopromat1
W=M1*(x(i)**2)/2.+R1*((x(i)-
a)**3*z(i,m))/6.+P1*(x(i)-
2*a)**3*z(i,2*m)/6.+&
P2*(x(i)-3*a)**3*z(i,3*m)/6.+P1*(x(i)-
4*a)**3*z(i,4*m)/6.+&
R2*(x(i)-5*a)**3*z(i,5*m)/6.+P3*(x(i)-
6*a)**3*z(i,6*m)/6.+&
q*(x(i)-a)**4*z(i,m)/24.-q*(x(i)-
4*a)**4*z(i,4*m)/24.
end

program sveta
use sopromat1
real(8):: V(nm), W2(nm), S(nm), F(nm)
real::D(2,2),C(2)

do j=1,n
if((j-1)*m>0)  x((j-1)*m+1)=x((j-1)*m)
do i=2,m
x((j-1)*m+i)=x((j-1)*m+i-1)+a/(m-1)
enddo
enddo

R2=(M1-5*q*a*2.5*a+2*q*a*4*a-P1*a-
P2*2*a-P1*3*a-P3*5*a)/(4*a)
R1=(-M1-5*q*a*1.5*a-P1*3*a-P2*2*a-
P1*a+P3*a)/(4*a)

Print*, R2,R1,x
D(1,:)=(/1.,a/);D(2,:)=(/1.,5*a/)
C=(/-1./EI*W(m),-1./EI*W(5*m)/)
Call SIMQ (D,C,2,ier)
Print*,C;
Do i=1,nm
V(i)=C(1)+C(2)*x(i)+W(i)/EI;
print*,i,v(i)
enddo
Call cls(2)
call grafic(10,10,410,110,x,V,nm)

Do i=1,nm
W2(i)=C(2)+(M1*(x(i)**1)/1.+R1*((x(i)
-a)**2*z(i,m))/2.+P1*(x(i)-
2*a)**2*z(i,2*m)/2.+&
P2*(x(i)-3*a)**2*z(i,3*m)/2.+P1*(x(i)-
4*a)**2*z(i,4*m)/2.+&
R2*(x(i)-5*a)**2*z(i,5*m)/2.+P3*(x(i)-
6*a)**2*z(i,6*m)/2.+&
q*(x(i)-a)**3*z(i,m)/6.-q*(x(i)-
4*a)**3*z(i,4*m)/6.)/EI
enddo
call grafic(10,120,410,220,x,W2,nm)

Do i=1,nm
S(i)=(M1+R1*(x(i)-a)*z(i,m)+P1*(x(i)-
2*a)*z(i,2*m)+&
P2*(x(i)-3*a)*z(i,3*m)+P1*(x(i)-
4*a)*z(i,4*m)+&
R2*(x(i)-5*a)*z(i,5*m)+P3*(x(i)-
6*a)*z(i,6*m)+&
q*(x(i)-a)**2*z(i,m)/2.-q*(x(i)-
4*a)**2*z(i,4*m)/2.)/EI
enddo
call grafic(10,230,410,330,x,S,nm)

Do i=1,nm
F(i)=(R1*z(i,m)+P1*z(i,2*m)+&
P2*z(i,3*m)+P1*z(i,4*m)+&
R2*z(i,5*m)+P3*z(i,6*m)+&
q*(x(i)-a)*z(i,m)-q*(x(i)-
4*a)*z(i,4*m))/EI
enddo
call grafic(10,340,410,440,x,F,nm)
end
```

Подпрограмма для построения эпюр:

```
Subroutine Cls(I)
Use dflib !подключение модуля, содержащего интерфейсы к графическим процедурам
Type(qwinfo)qv !описание величины qv, имеющей производный тип qwinfor. Тип данных qwinfor позволяет устанавливать параметры графического окна
Open(I,file='user') !открытие графического окна с номером I
qv. Type=2
result = SETWSIZEQQ(QWIN$FRAMEWINDOW, qv)
result =   SETWSIZEQQ(I, qv)
Result=setbkcolor(15) !устанавливает цвет фона графического экрана (белый)
call CLEARSCREEN(0)
Result=settextcolor(0)
Result=setbkcolor(7)
end

Subroutine grafic(jl,il,jr,ir,x,y,n)
Use dflib
Real(8)::x(n), y(n), yl, yr
Type(xycoord)xy
Type(wxycoord)wxy
!Moveto n1
Call setviewport(jl,il,jr,ir)
Result=setcolor(0)
!Result=rectangle(2,0,0,jr-jl,ir-il)
yl=max(0.,maxval(y)); yr=min(0.,minval(y))
Result=setwindow(.true.,x(1),yl,x(n),yr)
Do i=1,n-1; Call moveto_w(x(i),0d0,wxy)
Result=lineto_w(x(i),y(i))
Result=lineto_w(x(i+1),y(i+1))
Result=lineto_w(x(i+1),0d0)
Enddo
Call moveto_w(x(1),0d0,wxy)
Result=lineto_w(x(n),0d0)
End
```

Программа на Fortran'е достаточно объемна, она состоит из головной программы и подпрограммы для построения самих эпюр, так же требуется вызывать подпрограмму SIMQ, для решения СЛАУ.

Эпюры, построенные на Fortran'е (рис. 1.7) схожи с эпюрами, построенными в Mathcad'е.

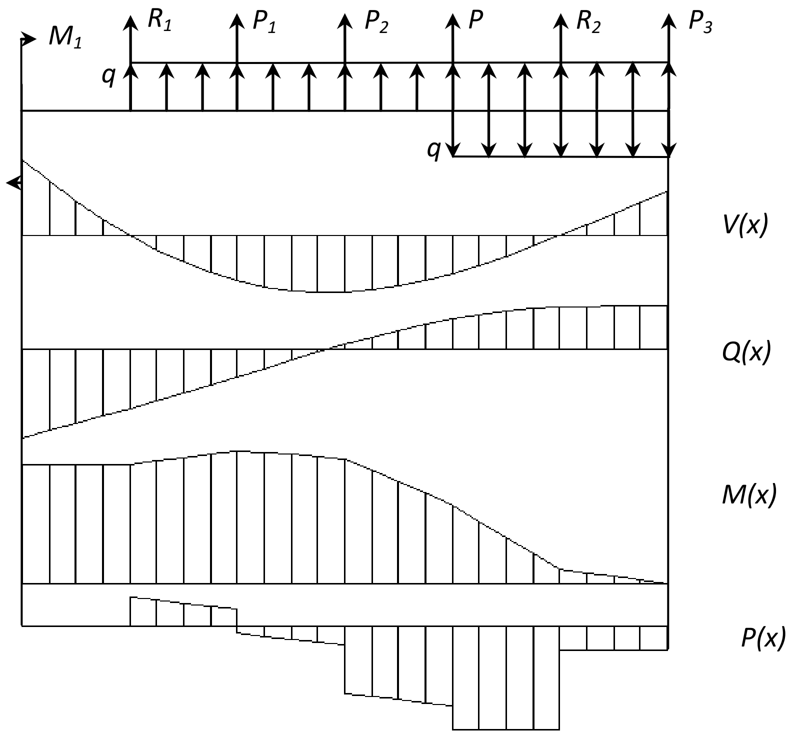


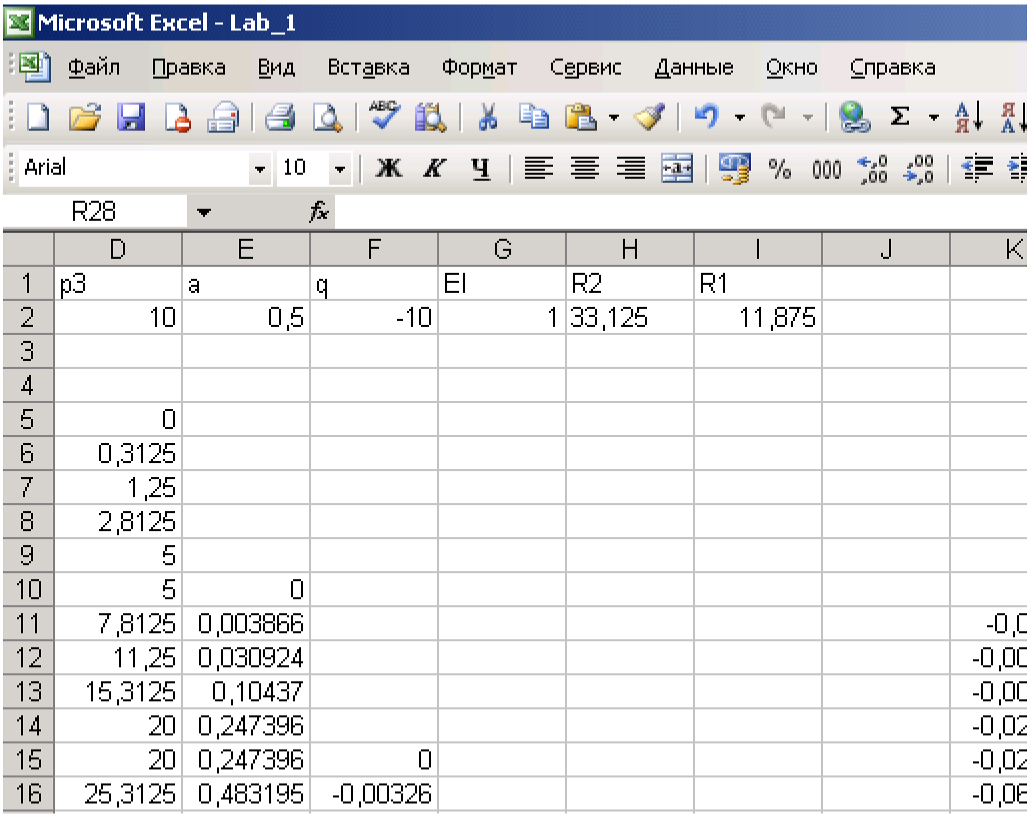
Рис.1.7 Эпюры построенные на Fortran'е

Различия между программами Mathcad и Fortran:

1. В Fortran'е не существует различия между заглавными и прописными буквами, а в Mathcad'е идет четкое различие ввода переменных.
2. Mathcad не строит эпюры, а строит графики в отличии от Fortran.
3. Для построения эпюр на Fortran'е мы должны составлять подпрограмму, в отличие от Mathcad'а, где нужно только назначить операцию построения графика.
4. Текст программы в Mathcad'е значительно легче и проще для усвоения.

Построение эпюр в Excel

В начале вводим исходные данные для внешней нагрузки. Затем задаем шаг сетки по *x* и номеруем точки, всего точек будет 30. Далее в другом столбце задаем значение для *x*, в первой строке пишем 0, во второй предыдущее значению плюс шаг, при этом повторяем значения для узловых точек. Задаем функции для внешних нагрузок и затем суммируем их. Далее задаем матрицы **D** и **B**, обращаем матрицу **D** и умножаем ее на **B** для получения неизвестных коэффициентов. Задаем уравнение упругой линии, в которое входят полученные коэффициенты. Далее выделяя столбец со значениями x и столбец уравнения упругой линии, строим эпюру перемещений (рис. 1.8).



	D	E	F	G	H	I	J	K
1	p3	a	q	EI	R2	R1		
2	10	0,5	-10	1	33,125	11,875		
3								
4								
5	0							
6	0,3125							
7	1,25							
8	2,8125							
9	5							
10	5	0						
11	7,8125	0,003866						-0,0
12	11,25	0,030924						-0,00
13	15,3125	0,10437						-0,00
14	20	0,247396						-0,02
15	20	0,247396	0					-0,02
16	25,3125	0,483195	-0,00326					-0,06

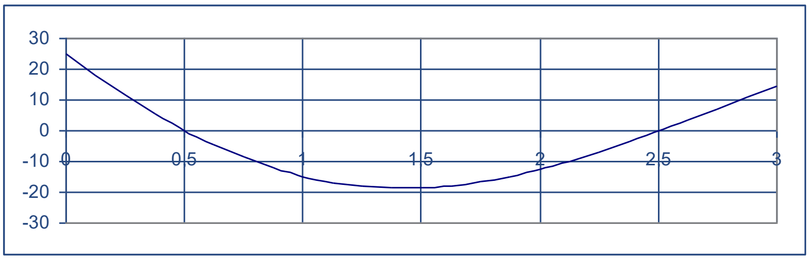


Рис.1.8 Эпюра прогибов балки Q(x)

Комплект заданий для зачета

по дисциплине *Практикум по алгоритмизации и программированию*
(наименование дисциплины)

Задача (задание) 1.Таблица футбольного чемпионата задана вектором X(10) (X(i)=i, где i- исходный номер команды) и матрицей A(10,10), где элемент A(i,j) численно равен количеству мячей, забитых i-ой командой j-ой. Постройте диаграмму результативности (количества забитых мячей) команд в зависимости от занятого места. Распечатайте таблицу распределения мест, набранных очков и результативность команд. Исходную матрицу A задайте самостоятельно.

Задача (задание) 2.Таблица футбольного чемпионата задана вектором X(10) (X(i)=i, где i- исходный номер команды) и матрицей A(10,10), где элемент A(i,j) численно равен количеству мячей, забитых i-ой командой j-ой. Найдите номера команд, имеющих больше побед, чем поражений. Постройте диаграмму результативности (количества забитых мячей) этих. Распечатайте эти команды, набранные ими очки и количество побед. Исходную матрицу A задайте самостоятельно.

Задача (задание) 3.Таблица футбольного чемпионата задана матрицей A(10,10) (A(i,i)=0, A(i,j)={0;1;2},i<>j) и вектором X(10) (X(i)=i, где i- исходный номер команды). Найдите номера команд, прошедших чемпионат без поражений. Постройте диаграмму результативности (количества забитых мячей) этих команд. Распечатайте эти команды, набранные ими очки и количество побед. Исходную матрицу A задайте самостоятельно.

Задача (задание) 4.Для целочисленной матрицы A(10,10) найдите матрицу из нулей и единиц B(10,10), элементы которой B(i,j) равны 1, если все соседи (элементы матрицы A, индексы которых отличаются от i,j не более чем на 1) меньше самого A(i,j). Организуйте графический вывод (в виде таблиц) результатов работы программы на экран. Исходную матрицу A задайте самостоятельно.

Задача (задание) 5.Для целочисленной матрицы A(10,10) найдите матрицу из нулей и единиц B(10,10), элементы которой B(i,j) равны 1, если среди соседей A(i,j) (элементов матрицы A, индексы которых отличаются от i,j не более чем на 1) есть не менее двух совпадающих с A(i,j). Организуйте графический вывод (в виде таблиц) результатов работы программы на экран. Исходную матрицу A задайте самостоятельно.

Задача (задание) 6.Дан целый массив A(20).Получите массив B(m) (динамический) из чисел,взятых по одному из каждой группы равных членов массива A. Постройте диаграмму количества одинаковых элементов в зависимости от их численных значений. Исходный массив A задайте самостоятельно.

Задача (задание) 7.Дан целый массив A(20). Получите массив B(m) (динамический), элементы которого равны всем целым в порядке возрастания из интервала [min(A(i)),max(A(i))], не входящим в A. Постройте диаграмму зависимости численных значений элементов массива B от порядкового номера. Исходный массив A задайте самостоятельно.

Задача (задание) 8.Найдите первые три корня a(i), i=1,2,3 нелинейного уравнения $ch(a)*cos(a)+1=0$.

Постройте графики трех функций

$$y(i)=U(a(i)*x) - S(a(i))/T(a(i))*V(a(i)*x), \quad i=1,2,3, \quad 0<=x<=1,$$

где S(x),T(x),U(x),V(x)- функции Крылова

Задача (задание) 9.Постройте графики трех функций $Y(i,x)=C(i)*W(i,x)$, i=1,2,3,

где: $W(i,x)=U(a(i)*x)-S(a(i))/T(a(i))*V(a(i)*x), \quad 0<=x<=1,$

$a(1)=1.875, a(2)=4.694, a(3)=5*pi/2$

$C(i)=sqrt(1/(\quad W(i,x)**2*dx)),$

S(x),T(x),U(x),V(x)- функции Крылова.

Задача (задание) 10.Найдите методом простых итераций min собственное значение d и соответствующий собственный вектор для системы линейных однородных уравнений $X=d*H*X$

где

$$H= \begin{Vmatrix} c1+c2+c5 & -c2 & -c5 \\ -c2 & c2+c3 & -c3 \\ -c5 & -c3 & c3+c4+c5 \end{Vmatrix}$$

Постройте диаграмму зависимости численных значений элементов собственного вектора от порядкового номера.

Вычисления выполните для следующих значений: c1=1, c2=1.8, c3=0.8, c4=2, c5=.5

Задача (задание) 11.Используя метод наименьших квадратов заданную табличную функцию

x:	-.5	1	2	3	4	5	6
y:	1	.2	0	.2	.5	1.3	2

аппроксимируйте многочленом третей степени $Q(x)=q0+q1*x+q2*x2+q3*x3$. На печать выдайте значения qi и Q(x) в заданных точках xi. Постройте график многочлена и на нем покажите заданные точки.

Задача (задание) 12.Дана функция комплексного переменного

Графически показать, в какую фигуру в плоскости z преобразуется окружность

Исследовать случаи n=1,2,10,100

Задача (задание) 13.Для области D в плоскости z=x+i*y D={ 0<arg(z)<pi/2, 1<|z|<2 } постройте область D* в плоскости w=u+i*v, отображенную функцией w=ln(i*z) (область D при обходе должна оставаться слева). Результат представьте в виде графиков.

Теоретические вопросы

1. Решение краевой задачи для дифференциального уравнения второго порядка методом прогонки

2. Аппроксимация функций

3. Машинная графика для построения эпюр

4. Методы решения задач на собственные значения

5. Математическая модель и программа решения задачи по теории колебаний.

6. Комплексные величины при выполнении расчетов

Критерии оценки

- Задание считается выполненным на **пороговом** уровне, если студент ориентируется в теоретический вопросах, оценка составляет 50 баллов
- Задание считается выполненным на **базовом** уровне, если студент отвечает на теоретический вопрос и допускает незначительную ошибку при решении задачи, оценка составляет 75 баллов
- Задание считается выполненным на **продвинутом** уровне, если студент отвечает на теоретический вопрос и решает задачу, оценка составляет 100 баллов

Зачет считается сданным, если средняя сумма баллов по всем заданиям составляет не менее 50 баллов (по 100 балльной шкале).
Коэффициент, с которым учитывается полученная сумма баллов в общей оценке по дисциплине, определяется Правилами аттестации.

Составитель ________________________ А.И. Белоусов
(подпись)

«____»________________20 г.