

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Основы алгоритмизации и программирования

: 15.03.03

, :

: 1, : 2

		2
1	()	4
2		144
3	, .	81
4	, .	0
5	, .	18
6	, .	54
7	, .	0
8	, .	2
9	, .	7
10	, .	63
11	(, ,)	
12		

(): 15.03.03

220 12.03.2015 ., : 16.04.2015 .

: 1,

(): 15.03.03

, 5/1 20.06.2017

, 5 21.06.2017

:

,

:

,

:

. . . .

1.

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:	
5.	
6.	
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:	
2.	
3.	
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:	
1.	
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:	
1.	,
3.	

2.

2.1

(
---	--

.4. 2	
1.знать структуру и область применения современных языков программирования	;
.4. 3	
2.уметь программировать на одном из языков высокого уровня	;
.5. 1	
3.уметь работать со специальными программами для составления отчетов и презентаций	;
.6. 1	
4.знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей	;

.6. 3	
5.знать основные специализированные программные средства для решения профессиональных задач	;
.4. 3	
6.уметь использовать языки и системы программирования для решения профессиональных задач	;
.10. 5	
7.уметь использовать специализированные программные средства при решении профессиональных задач	;
.10. 6	
8.уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	;

3.

3.1

	,	.		
: 2				
:				
1.	0	4	2, 5, 6, 7, 8	(Excel, MathCAD, FORTRAN)
2.	0	12	2, 3, 6, 7, 8	(Excel, MathCAD, FORTRAN)
3.	0	12	2, 3, 6, 7, 8	(Excel, MathCAD, FORTRAN)
4.	0	12	2, 3, 6, 7, 8	(Excel, MathCAD, FORTRAN)

5.	0	6	2, 4, 5	(Excel, MathCAD, FORTRAN)
6.	0	4	2, 4, 5	(Excel, MathCAD, FORTRAN)
7.	0	4	2, 4, 5	(Excel, MathCAD, FORTRAN)

3.2

	,			
: 2				
:				
1.	0	2	2, 5, 7, 8	
2.	0	2	1, 4, 5	
3.	0	4	1, 4, 5	
4.	0	4	1, 4, 5	
5.	0	2	3, 4	

6.	0	2	2, 4, 5	.
7.	0	2	2, 3, 4, 5	.

4.

: 2				
1		1, 2, 3, 4, 5, 6, 7, 8	40	0
, 10 4 5 6 7 8 9 : " " 1- / . . . - ;[.: . . .]. - , 2008. - 54, [2] .: ., .. - : http://elibrary.nstu.ru/source?bib_id=vtls000082444				
2		1, 2, 3, 4, 5, 6, 7, 8	18	0
: " " 1- : / . . . - ;[.: . . .]. - , 2008. - 54, [2] .: ., .. - : http://elibrary.nstu.ru/source?bib_id=vtls000082444				
3		1, 2, 3, 4, 5, 6, 7, 8	5	7
: " " 1- : / . . . - ;[.: . . .]. - , 2008. - 54, [2] .: ., .. - : http://elibrary.nstu.ru/source?bib_id=vtls000082444				

5.

- , (. 5.1).

5.1

	-
	e-mail;
	e-mail;
	e-mail;
	e-mail; ;

6.

(),

-
15-

ECTS.

. 6.1.

6.1

: 2	
Курсовая работа: Итого	80
Зачет:	20

6.2

6.2

		/	
.10	5.	+	
	6.	+	
.4	2.		+
	3.	+	
.5	1.	+	
.6	1. , ,	+	+
	3.		+

1

7.

1. Гаврилов М. В. Информатика и информационные технологии : учебник для бакалавров / М. В. Гаврилов, В. А. Климов. - Москва, 2012. - 349, [1] с. : табл., ил.
2. Краинский И. Word 2007 : популярный самоучитель / И. Краинский. - СПб. [и др.], 2008. - 240 с. : ил.
3. Бартенев О. В. Современный Фортран / О. В. Бартенев. - М., 2000. - 448 с.

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.

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. Плис А. И. Лабораторный практикум по высшей математике : учебное пособие для втузов / А. И. Плис, Н. А. Сливина. - М., 1983. - 208 с.
3. Острейковский В. А. Информатика : Учеб. для вузов. - М., 2001. - 511 с.
4. Бартеньев О. В. Современный Фортран : [учебное пособие] / О. В. Бартеньев. - М., 1998. - 397 с. : табл.

-

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>
5. :

8.

8.1

1. Информатика и программирование : методические указания для проведения лабораторных работ по курсу "Информатика" для 1-ого курса ФЛА / Новосиб. гос. техн. ун-т ; [сост.: М. А. Леган и др.]. - Новосибирск, 2008. - 54, [2] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000082444
2. Численные методы и программирование на Фортране : методические указания для проведения лабораторных работ для 1-2 курсов факультета летательных аппаратов / Новосиб. гос. техн. ун-т ; [сост.: Н. В. Третьякова, П. Е. Рябчиков]. - Новосибирск, 2010. - 43, [1] с. : ил., табл.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2010/2010_3917.pdf

8.2

- 1 Microsoft Office
- 2 Microsoft Windows

9.

-

1	(- , ,)	

--	--	--

1	(,
	Internet)	

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

Основы алгоритмизации и программирования

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

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Основы алгоритмизации и программирования 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	у5. уметь использовать специализированные программные средства при решении профессиональных задач	Вычисление длины дуги кривой линии Основы алгоритмизации и программирования. Расчет длины дуги кривой линии Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных уравнений	Курсовая работа, разделы 1-5	
ОПК.10	у6. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	Вычисление длины дуги кривой линии Основы алгоритмизации и программирования. Расчет длины дуги кривой линии Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных уравнений	Курсовая работа, разделы 1-5	
ПК.4/НИ 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	з2. знать структуру и область применения современных языков программирования	Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных алгебраических уравнений		Зачет, вопросы 1-10

экспериментально о оборудования для проведения механических испытаний				
ПК.4/НИ	у3. уметь программировать на одном из языков высокого уровня	Вычисление длины дуги кривой линии Основы алгоритмизации и программирования. Расчет длины дуги кривой линии Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных уравнений	Курсовая работа, разделы 1-5	
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	Оформление курсовой работы Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных уравнений	Курсовая работа, разделы 1-5	
ПК.6/НИ 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	з1. знать современные программные средства оформления отчетов, презентаций, рефератов, докладов и статей	Оформление курсовой работы Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных алгебраических уравнений	Курсовая работа, разделы 1-5	Зачет, вопросы 1-10

ПК.6/НИ	33. знать основные специализированные программные средства для решения профессиональных задач	Вычисление длины дуги кривой линии Основы алгоритмизации и программирования. Расчет длины дуги кривой линии Решение обыкновенных дифференциальных уравнений. Задача Коши Решение систем линейных алгебраических уравнений Решение систем нелинейных алгебраических уравнений		Зачет, вопросы 1-10
---------	---	--	--	---------------------

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 2 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.

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

В 2 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

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 не достаточно, 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.

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

Паспорт зачета

по дисциплине «Основы алгоритмизации и программирования», 2 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-5, второй вопрос из диапазона вопросов 6-10 (список вопросов приведен ниже). 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 «Основы алгоритмизации и программирования»

1. Алгоритм вычисления длины дуги кривой.
2. Решение задачи Коши. Алгоритм метода Эйлера.
3. Методом Эйлера найдите решение уравнения на отрезке $[0,1]$ $y'''=2*x**2*y''-3*x*y'+4*y-\sin(x)$ при $y(0)=1, y'(0)=0, y''(0)=0$.

Утверждаю: зав. кафедрой ПЛА _____ проф., Пустовой Н.В.
(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 показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки, оценка составляет *25 баллов*.
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает не принципиальные ошибки, например, вычислительные, оценка составляет *50 баллов*.
- Ответ на билет для зачета билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, не допускает ошибок при решении задачи,

оценка составляет 75 баллов.

- Ответ на билет для зачета билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет 100 баллов.

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем заданиям билета оставляет не менее 59 баллов (из 100 возможных).

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 «Основы алгоритмизации и программирования»

- 1) Алгоритм вычисления длины дуги кривой.
- 2) Решение нелинейных уравнений. Алгоритм метода половинного деления.
- 3) Решение нелинейных уравнений. Алгоритм метода Ньютона.
- 4) Решение нелинейных уравнений. Алгоритм метода простых итераций.
- 5) Решение систем линейных алгебраических уравнений. Алгоритм метода простых итераций.
- 6) Определение экстремумов функции. Алгоритм метода золотого сечения.
- 7) Решение задачи Коши. Алгоритм метода Эйлера.
- 8) Решение задачи Коши. Алгоритм метода Эйлера-Коши.
- 9) Решение задачи Коши. Алгоритм метода Рунге-Кутты 4-го порядка точности.
- 10) Фортран. Разработка графических программ. Установка видеорежима. Графические координаты. Формирование графического изображения. Вывод графического текста.

5. Задачи к зачету по дисциплине «Основы алгоритмизации и программирования»

Задача 1. Методом Эйлера найдите решение уравнения на отрезке $[0,1]$ $y'''=2*x**2*y"-3*x*y'+4*y-\sin(x)$ при $y(0)=1, y'(0)=0, y''(0)=0$.

Задача 2. На отрезке $[0,1]$ методом Эйлера-Коши найдите решение уравнения $y'=\sin(x*y)$ при $y(0)=1$.

Задача 3. Решите систему уравнений $A*X=B$

Задача 4. Используя стандартную подпрограмму MINV, найдите решение СЛАУ

Задача 5. Решите методами Эйлера и Рунге-Кутты дифференциальное уравнение третьего порядка на отрезке $[0,1]$ при начальных условиях $y(1)=1, y'(1)=1, y''(1)=-.5$. Постройте графики полученных решений.

Задача 6. Постройте графики и найдите все точки пересечения двух окружностей:
 $x**2+y**2=9$ $(x-3.5)**2+(y-2)**2=9$

Задача 7. Найдите решение В системы линейных алгебраических уравнений где: $z=\{0.2, 0.3, 0.4\}$. Постройте график функции на отрезке $[0, 1]$.

Задача 8. Постройте график и на нем укажите все экстремумы функции $y(x)$ на участке $.2 \leq x \leq 1.2$ $y(x)=1.2*\sin(\pi*x)-.8*\sin(3*\pi*x)+1.4*\sin(5*\pi*x)$

Паспорт курсовой работы

по дисциплине «Основы алгоритмизации и программирования», 2 семестр

1. Методика оценки.

Задание: решить поставленные задачи, оформив их решение по правилам оформления.

Структура: титульный лист, содержание/оглавление, введение, основная часть, заключение, список литературы, приложения (при необходимости).

Этапы выполнения и защиты: 1) составление алгоритмов решения задач, 2) написание и отладка программ, 3) решение задач, 4) оформление результатов в соответствии с требованиями, 5) ответы на вопросы при защите.

Оцениваемые позиции: правильность решения задач, правильность оформления (в том числе качество доведения информации до проверяющего КР), ответы на вопросы при защите.

Требования к оформлению: КР выполняется с помощью текстового процессора (желательно MS Word), параметры страницы: поля – левое – 2.5см., правое – 1см., верхнее и нижнее – 1.5см. Текст – 12пт., Times New Roman, межстрочный интервал – одинарный, выравнивание абзацев – по ширине, рисунки не должны иметь чёрный/тёмный фон, заполненность страницы информацией не должна быть менее 70% (без необходимости); главы, разделы, подразделы нумеруются арабскими цифрами (разделитель – точка); каждая новая глава (заголовок первого уровня) начинается с новой страницы, рисунки должны иметь подпись с номером и названием, ссылку на себя выше по тексту; таблицы должны иметь надпись с номером и иметь ссылку на себя выше по тексту (название таблиц не обязательно).

2. Критерии оценки.

- работа считается **не выполненной**, если не решена хотя бы одна задача, оценка составляет 25 баллов.
- работа считается выполненной **на пороговом** уровне, если КР содержит описание и решение всех задач, но допущено более двух ошибок при решении задач или оформлении, оценка составляет 50 баллов.
- работа считается выполненной **на базовом** уровне, если КР выполнена по всем требованиям, содержит описание и решение всех задач, но допущена одна ошибка, оценка составляет 75 баллов.
- работа считается выполненной **на продвинутом** уровне, если КР оформлена по всем правилам, задачи решены верно, оценка составляет 100 баллов.

3. Шкала оценки.

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

4. Примерный перечень тем курсового проекта (работы).

Тема: Алгоритмизация вычислительных методов

Задания:

- 1) Решение нелинейных уравнений.
- 2) Решение систем линейных алгебраических уравнений.
- 3) Решение задачи коши.
- 4) Решение систем нелинейных уравнений.
- 5) Разложение функций в ряд Фурье.
- 6) Определение экстремумов функции методом золотого сечения.
- 7) Определение собственных частот и форм колебаний вала.

5. Перечень вопросов к защите курсового проекта (работы).

- 1) Алгоритмы методов решения нелинейных уравнений.
- 2) Алгоритмы методов решения систем линейных алгебраических уравнений.
- 3) Алгоритмы методов решения задачи Коши.
- 4) Алгоритмы методов решения систем нелинейных уравнений.
- 5) Алгоритм разложение функций в ряд Фурье.
- 6) Алгоритм определение экстремумов функции методом золотого сечения.
- 7) Алгоритм определения собственных частот и форм колебаний вала.

Задача 7. ВЫЧИСЛЕНИЕ ДЛИНЫ КРИВОЙ.

На отрезке $[a, b]$ вычислить длину кривой S для функции $y=f(x)$:

1) аппроксимируя кривую ломаной линией и вычисляя длину этой линии

$$S = \sum_{i=1}^n s_i, \quad n - \text{число разбиений, } s_i - \text{длина прямой, соединяющей две точки } (x_i, f(x_i)) \text{ и } (x_{i+1}, f(x_{i+1})),$$

$$s_i = \sqrt{[f(x_{i+1}) - f(x_i)]^2 + h^2}, \quad h = (b-a)/n, \quad x_i = a + (i-1) \cdot h$$

2) используя формулу $S = \int_a^b \sqrt{1 + [f'(x)]^2} dx$

Интеграл вычислить методами : а)трапеций; б)Симпсона; в)Гаусса.

Исследовать точность вычисления S способами 1), 2) в зависимости от числа разбиений отрезка $[a, b]$.

Варианты заданий.

N	f(x)	a	b	N	f(x)	a	b
1	$\sin(x)$	0	π	11	$x \cdot \sin(x)$	0	$\pi/3$
2	$\sin(x)^2$	0	$\pi/4$	12	$x \cdot \cos(x)$	$\pi/2$	π
3	$\sin(x^2)$	0	$\pi/3$	13	$\exp(x)$	-1	1
4	$\cos(x)$	0	$\pi/2$	14	$x \cdot \exp(x)$	-.3	1.3
5	$\cos(x)^2$	$\pi/2$	π	15	$\cos(3 \cdot x - 1)$.51	$\pi/3$
6	$\cos(x^2)$	$\pi/3$	$\pi/2$	16	$x^2 - \cos(x)$.1	.75
7	$\sin(x)/x$.1	$\pi/2$	17	$x - \sin(2 \cdot x)$.3	.9
8	$\sin(x + x^3)$.7	2	18	$4 \cdot x^3 - x$.7	1.3
9	$\cos(\sin(x))$	0	$\pi/2$	19	$2 \cdot x^3 - x^2$	2.5	3.1
10	$\sin(\cos(x))$	$\pi/2$	π	20	$x - \cos(x)^2$	0	$\pi/2$

Итерации продолжаются до тех пор, пока все $x_k^{(i+1)}$ не станут достаточно близки к $x_k^{(i)}$. Построенный по схеме (6) итерационный процесс сходится, если коэффициенты исходной матрицы A удовлетворяют условию

$$|a_{ii}| > \sum_{j=1,2,\dots,j \neq i}^n |a_{ij}|, \quad i=1,2,\dots,n, \quad (6)$$

т.е. если в каждой строке модуль диагонального элемента больше суммы модулей остальных элементов строки. Если коэффициенты решаемой системы не удовлетворяют условию сходимости, то с помощью линейного комбинирования уравнений исходной системы (в простейшем случае - перестановкой) получают новую, для которой условие (7) выполняется. Например, пусть дана система уравнений

$$1 \cdot x_1 + 6 \cdot x_2 + 2 \cdot x_3 - 1 \cdot x_4 = 2 \quad (a)$$

$$3 \cdot x_1 - 1 \cdot x_2 + 7 \cdot x_3 - 2 \cdot x_4 = -3 \quad (b)$$

$$-8 \cdot x_1 + 1 \cdot x_2 - 4 \cdot x_3 + 1 \cdot x_4 = 4 \quad (c)$$

$$4 \cdot x_1 - 2 \cdot x_2 + 1 \cdot x_3 + 9 \cdot x_4 = 0 \quad (d)$$

Новую систему получим, если в качестве первого уравнения запишем (c), второго - (a), третьего - (b) и четвертого - (d). Такое преобразование СЛАУ можно выполнить как вручную, так и программно, переставляя соответствующим образом строки матрицы A и элементы вектора B .

Выбор начальных приближений не влияет на сходимость метода, поэтому можно положить $x_i=0$ или $x_i=d_i$.

При оформлении метода простых итераций в виде подпрограммы в качестве формальных параметров можно принять: $C, D, X, Y, N, EPS, MMAX, M, IER$.

Входные параметры:

C - массив $N \times N$ - матрица коэффициентов системы (3);

D - одномерный массив из N элементов - вектор системы (3);

N - порядок системы;

EPS - точность вычисления корней системы;

$MMAX$ - предельно допустимое количество итераций.

Выходные параметры:

X - массив из N элементов - корни системы (3);

M - количество проведенных итераций;

IER - код ошибки ($IER=0$ - решение найдено, $IER=1$ - количество итераций превысило предельно допустимое).

Рабочие параметры:

Y - массив из N элементов.

Алгоритм метода:

- 1) Задаются: начальное значение количества итераций $M=0$; код ошибки $IER=1$; начальное приближение для X ($x_i=d_i$, $i=1,...,n$). Далее выполняется итерационный цикл:
- 2) Счетчик итераций изменяется на единицу $M=M+1$ и проверяется условие $M>M_{MAX}$. Если условие выполняется, то осуществляется возврат в вызывающую программу с кодом ошибки $IER=1$, в противном случае процесс продолжается.
- 3) В двойном цикле вычисляется последующее приближение $Y=C*X+D$.
- 4) Определяется максимальный модуль разности x_i и y_i : $s=\max(|x_i - y_i|)$.
- 5) Задаются значения X для выполнения следующего приближения: $x_i=y_i$.
- 6) Выполняется проверка на выход из итерационного цикла: если $s<EPS$, то изменяется код ошибки $IER = 0$ и осуществляется возврат в вызывающую программу, иначе цикл повторяется с пункта 2.

ЗАДАНИЕ. Решить СЛАУ $A*X=B$ (см. варианты заданий) методами:

- 1) Гаусса, используя стандартную подпрограмму SIMQ;
- 2) обратной матрицы по формуле $X = A^{-1} * B$, используя для обращения матрицы стандартную подпрограмму MINV;
- 3) простых итераций, составив собственную подпрограмму метода и подпрограмму вычисления массивов C и D по формулам (4).

Преобразование заданной системы уравнений (массивов A и B) так, чтобы выполнялось условие сходимости, выполнить либо вручную, либо - по заданию преподавателя - программно. Исходные данные вводить из файла данных, результаты печатать в этот же файл.

ИСПОЛЬЗУЕМЫЕ стандартные подпрограммы.

Подпрограмма решения СЛАУ методом Гаусса: SIMQ (A,B,N,IER).

Входные параметры:

A(N,N) - матрица коэффициентов системы уравнений (2);

B(N) - массив свободных членов в уравнении (2);

N - порядок системы.

Выходные параметры:

B(N) - корни системы (2);

IER - код ошибки: IER=0 - решение получено; IER=1 - матрица A - вырожденная.

При работе SIMQ матрица A портится.

Подпрограмма обращения матрицы: MINV (A,N,D,L,M).

Входные параметры:

A(N,N) - исходная матрица;

N - порядок системы.

Выходные параметры:

A(N,N) - обратная матрица;

D - определитель матрицы A.

Рабочие параметры:

L(N),M(N) - используемые при вычислении обратной матрицы массивы,

должны быть описаны в вызывающей программе.

ВАРИАНТЫ ЗАДАНИЙ.

1	$\begin{vmatrix} 1 & 8 & -2 & 3 & -1 \\ 6 & 2 & 0 & 3 & 4 \\ -1 & 4 & 6 & 0 & 3 \\ 3 & -2 & 1 & -9 & -7 \end{vmatrix}$	2	$\begin{vmatrix} 3 & 9 & -1 & 4 & 6 \\ -1 & 2 & 8 & 3 & -4 \\ -1 & -2 & 2 & 6 & -7 \\ 5 & -2 & 1 & 0 & 1 \end{vmatrix}$	3	$\begin{vmatrix} 0 & 3 & -7 & -2 & -2 \\ 1 & 2 & 0 & 6 & 3 \\ -8 & 4 & 2 & 0 & -3 \\ 1 & -4 & 1 & 0 & 7 \end{vmatrix}$	4	$\begin{vmatrix} 1 & 6 & -2 & 2 & -1 \\ 9 & 2 & 0 & 3 & 4 \\ -1 & 3 & 2 & 7 & 0 \\ 3 & -2 & 8 & -2 & -5 \end{vmatrix}$
5	$\begin{vmatrix} -1 & 7 & 3 & 2 & 5 \\ -8 & -2 & 1 & 3 & -6 \\ -1 & 2 & 3 & 7 & 8 \\ 3 & -2 & -8 & -2 & 1 \end{vmatrix}$	6	$\begin{vmatrix} 1 & -3 & -6 & 0 & -4 \\ 1 & 2 & -1 & 6 & 5 \\ -8 & 4 & 2 & -1 & -3 \\ 1 & -4 & 1 & 0 & -7 \end{vmatrix}$	7	$\begin{vmatrix} 0 & -2 & 3 & 6 & -1 \\ 6 & -1 & 1 & -3 & 5 \\ -1 & 7 & 2 & -2 & 3 \\ 3 & -2 & 7 & -1 & 2 \end{vmatrix}$	8	$\begin{vmatrix} -2 & -9 & 1 & 4 & 3 \\ -1 & -2 & 8 & 3 & -9 \\ -1 & -2 & 2 & 6 & -7 \\ 5 & -2 & 1 & 0 & 1 \end{vmatrix}$
9	$\begin{vmatrix} 3 & 1 & -2 & 7 & -1 \\ 7 & -2 & 0 & -3 & -3 \\ 1 & -6 & 2 & 1 & 1 \\ -3 & 2 & 9 & 1 & -4 \end{vmatrix}$	10	$\begin{vmatrix} 2 & 8 & -1 & -3 & 2 \\ -2 & 1 & 7 & 3 & 4 \\ 1 & -2 & -2 & 8 & -5 \\ 6 & -2 & 1 & -1 & 3 \end{vmatrix}$	11	$\begin{vmatrix} 1 & -3 & -7 & 2 & -1 \\ 0 & 2 & 2 & -6 & 4 \\ -8 & -4 & 2 & 0 & 3 \\ -1 & -4 & 1 & 0 & -6 \end{vmatrix}$	12	$\begin{vmatrix} -1 & 6 & 0 & 2 & -6 \\ 8 & -2 & 1 & 3 & 1 \\ -2 & 1 & -2 & 6 & 2 \\ -3 & 2 & -8 & -2 & 5 \end{vmatrix}$
13	$\begin{vmatrix} -1 & 6 & -1 & 2 & -4 \\ 8 & -2 & 1 & 3 & 5 \\ -2 & 1 & -2 & 7 & 1 \\ 3 & 0 & 8 & -2 & -3 \end{vmatrix}$	14	$\begin{vmatrix} 1 & 0 & -7 & -2 & -3 \\ -1 & 2 & 1 & 5 & -5 \\ -7 & 3 & 1 & -1 & -2 \\ 1 & -4 & -1 & 0 & 6 \end{vmatrix}$	15	$\begin{vmatrix} 2 & 1 & -2 & 7 & -1 \\ 6 & 1 & -2 & 1 & 2 \\ -1 & 4 & -2 & 0 & 4 \\ -3 & 1 & 6 & 0 & 3 \end{vmatrix}$	16	$\begin{vmatrix} -3 & 9 & 1 & -4 & -5 \\ 1 & -2 & 7 & -3 & -3 \\ -1 & 0 & 2 & -6 & -5 \\ 8 & -2 & 1 & 4 & 1 \end{vmatrix}$
	$\begin{vmatrix} 2 & -3 & -2 & 8 & -4 \end{vmatrix}$	18	$\begin{vmatrix} 2 & 8 & -1 & -4 & 1 \end{vmatrix}$	19	$\begin{vmatrix} 1 & 2 & -7 & -3 & -4 \end{vmatrix}$	20	$\begin{vmatrix} 0 & -6 & 3 & 2 & -7 \end{vmatrix}$

$$17 \left\| \begin{array}{ccccc} -7 & 2 & 2 & 1 & 5 \\ 1 & -4 & -1 & 1 & 2 \\ -3 & -1 & 9 & 0 & -6 \end{array} \right\| \left\| \begin{array}{ccccc} -1 & -3 & 9 & 2 & -2 \\ 0 & -2 & -2 & 6 & -4 \\ 5 & -2 & 1 & 1 & -1 \end{array} \right\| \left\| \begin{array}{ccccc} -1 & 2 & 0 & -6 & 9 \\ -7 & 4 & -2 & 0 & -8 \\ 1 & -4 & 1 & 0 & -7 \end{array} \right\| \left\| \begin{array}{ccccc} 9 & -2 & 0 & 3 & -4 \\ 1 & 3 & 0 & -7 & 0 \\ 1 & -2 & 8 & -2 & 5 \end{array} \right\|$$

9. РЕШЕНИЕ ЗАДАЧИ КОШИ.

Задача Коши формулируется следующим образом: на отрезке $[x_0, x_k]$ требуется найти решение $y=y(x)$ обыкновенного дифференциального уравнения $dy/dx=f(x,y)$, удовлетворяющее начальному условию $y(x_0)=y_0$. Численное решение задачи состоит в построении таблицы приближенных значений $y_1, y_2, y_3, \dots, y_m$ решения уравнения $y(x)$ в точках x_1, x_2, \dots, x_m . Обычно принимают $x_i = x_0 + i \cdot h$, $i=1, 2, \dots, m$. Точки x_i называются узлами сетки, а величина h - шагом: $h=(x_k-x_0)/m$. Существуют два широких класса методов приближенного решения задачи Коши.

1. Одноступенчатые (одношаговые) методы, в которых для нахождения следующей точки $y(x_{i+1})$ требуется информация лишь об одном предыдущем шаге. Сюда относятся метод решения с помощью рядов Тейлора, а также методы Рунге-Кутты различного порядка точности.

2. Многоступенчатые (многошаговые) методы, в которых для отыскания следующей точки решения $y(x_{i+1})$ используется информация более чем об одной из предыдущих точек и для достижения достаточной точности требуются итерации. Большинство методов этого класса называются методами прогноза и коррекции (методы Милна, Адамса и др.). Далее рассматриваются только одношаговые методы.

Разложение решения $y=y(x)$ в окрестности точки x в ряд Тейлора имеет вид $y_{i+1} =$

$$y_i + h \cdot y_i' + \frac{h^2}{2!} y_i'' + \frac{h^3}{3!} y_i''' + \dots,$$

где $y_{i+1}=y(x_i)$. Чем больше членов ряда используется для вычислений, тем точнее будет приближение. С точки зрения практического использования этот метод неудобен, так как необходимо вычислять различные производные $y(x)$. Однако он используется как критерий, показывающий, насколько тот или иной практически применяемый метод согласуется с разложением в ряд Тейлора. С этой целью вводится понятие порядка точности метода: если с помощью какого-либо приближенного метода удастся описать $n+1$ член разложения функции $y=y(x)$ в ряд Тейлора, то говорят, что метод n -го порядка точности (в разложении в ряд Тейлора учитываются члены до порядка h^n включительно). В методах Рунге-Кутты значение функции y_{i+1} в точке x_{i+1} определяется по формуле

$$y_{i+1} = y_i + \Delta y_i.$$

Наиболее простым является метод Рунге-Кутты первого порядка точности (или метод Эйлера), в котором для вычисления Δy_i используется касательная только в точке (x_i, y_i) :

$$\Delta y_i = h \cdot f(x_i, y_i).$$

Этот метод имеет довольно большую ошибку ограничения и часто оказывается неустойчивым - ошибка увеличивается с ростом x . В исправленном методе Эйлера (метод

Эйлера-Коши), который относится к семейству методов Рунге-Кутты второго порядка точности, вычисляется средний тангенс угла наклона касательной для двух точек (x_i, y_i) , $(x_{i+1}, y_i + h \cdot f(x_i, y_i))$:

$$\Delta y_i = (k_1 + k_2)/2, \quad k_1 = h \cdot f(x_i, y_i), \quad k_2 = h \cdot f(x_{i+1}, y_i + k_1).$$

Одним из самых употребительных методов интегрирования дифференциальных уравнений является метод Рунге-Кутты четвертого порядка точности (или просто метод Рунге-Кутты без указания точности) :

$$\begin{aligned} \Delta y_i &= (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)/6, \\ k_1 &= h \cdot f(x_i, y_i), & k_2 &= h \cdot f(x_i + h/2, y_i + k_1/2), \\ k_3 &= h \cdot f(x_i + h/2, y_i + k_2/2), & k_4 &= h \cdot f(x_{i+1}, y_i + k_3). \end{aligned}$$

Методы Рунге-Кутты переносятся на нормальные системы дифференциальных уравнений

$$y_j'(x) = f_j(x, y_1, y_2, \dots, y_n), \quad j=1, 2, \dots, n,$$

которые удобно записывать в векторной форме

$$Y'(x) = F(x, Y),$$

где $Y = (y_1, y_2, \dots, y_n)$, $F = (f_1, f_2, \dots, f_n)$. В расчетных формулах методов Рунге-Кутты y и $f(x, y)$ заменяется соответственно на Y и $F(x, Y)$, коэффициенты k_l - на K_l ($K_l = (k_{l1}, k_{l2}, \dots, k_{ln})$), и т.д..

При решении обыкновенного дифференциального уравнения n -го порядка последнее сводится к системе из n уравнений первого порядка. Например, из уравнения

$$y'' = f(x, y, y')$$

с начальными условиями $y(x_0) = y_0, y'(x_0) = p_0$,

заменой $y_1 = y, y_2 = y'$

получим систему $y_1' = y_2$,

$$y_2' = f(x, y_1, y_2)$$

с начальными условиями $y_1(x_0) = y_0, y_2(x_0) = p_0$

или в векторной форме $Y' = F(x, Y), Y(x_0) = Y_0$,

где $Y = (y_1, y_2), F = (f_1, f_2) = (y_2, f), f = f(x, y_1, y_2), Y_0 = (y_0, p_0)$.

Формулы метода Рунге-Кутты для системы дифференциальных уравнений второго порядка имеют вид (в формулах индекс i при y_{1i}, y_{2i} опущен).

(в общем случае)

(для уравнения 2-го порядка, сведенного к системе 2-х уравнений)

$$k_{l1} = h \cdot f_l(x_i, y_1, y_2) = h \cdot y_2,$$

$$\begin{aligned}
k_{12} &= h * f_2(x_i, y_1, y_2) &= h * f(x_i, y_1, y_2), \\
k_{21} &= h * f_1(x_i + h/2, y_1 + k_{11}/2, y_2 + k_{12}/2) &= h * f(x_i + h/2, y_1 + k_{11}/2, y_2 + k_{12}/2), \\
k_{22} &= h * f_2(x_i + h/2, y_1 + k_{11}/2, y_2 + k_{12}/2) &= h * f(x_i + h/2, y_1 + k_{11}/2, y_2 + k_{12}/2), \\
k_{31} &= h * f_1(x_i + h/2, y_1 + k_{21}/2, y_2 + k_{22}/2) &= h * f(x_i + h/2, y_1 + k_{21}/2, y_2 + k_{22}/2), \\
k_{32} &= h * f_2(x_i + h/2, y_1 + k_{21}/2, y_2 + k_{22}/2) &= h * f(x_i + h/2, y_1 + k_{21}/2, y_2 + k_{22}/2), \\
k_{41} &= h * f_1(x_{i+1}, y_1 + k_{31}, y_2 + k_{32}) &= h * f(x_{i+1}, y_1 + k_{31}, y_2 + k_{32}), \\
k_{42} &= h * f_2(x_{i+1}, y_1 + k_{31}, y_2 + k_{32}) &= h * f(x_{i+1}, y_1 + k_{31}, y_2 + k_{32}),
\end{aligned}$$

$$\Delta y_{1i} = (k_{11} + 2 * k_{21} + 2 * k_{31} + k_{41}) / 6, \quad \Delta y_{2i} = (k_{12} + 2 * k_{22} + 2 * k_{32} + k_{42}) / 6,$$

$$y_{1(i+1)} = y_{1i} + \Delta y_{1i}, \quad y_{2(i+1)} = y_{2i} + \Delta y_{2i}.$$

Аналогично можно записать формулы методов Эйлера и Эйлера-Коши.

ЗАДАНИЕ. Решить задачу Коши для дифференциального уравнения второго порядка методами Эйлера, Эйлера-Коши, Рунге-Кутты и сравнить с точным решением.

РЕКОМЕНДАЦИИ. Решение задачи Коши методами Эйлера, Эйлера-Коши, Рунге-Кутты необходимо оформить в виде подпрограмм. В подпрограммах интегрирование может выполняться либо на всем отрезке $[x_0, x_k]$ (выходными данными являются массивы из m значений y_1, y_2), либо только на одном шаге. Подпрограммы могут быть оформлены так, чтобы они могли применяться либо для любой системы дифференциальных уравнений, либо только для решения конкретной задачи. В последнем случае в качестве параметров каждой подпрограммы (при интегрировании на одном шаге) можно использовать: F, Y, X, H.

Входные параметры:

F - имя подпрограммы-функции, вычисляющей значение $f(x, y_1, y_2)$

Y(2)- одномерный массив - значения y_1, y_2 в точке x_i ;

X - текущее значение x_i ;

H - шаг интегрирования h .

Выходные параметры:

Y(2) - значения y_1, y_2 в точке x_{i+1} .

В теле подпрограммы в зависимости от метода могут использоваться внутренние массивы, которые, как и массив Y(2), должны быть описаны. Например, для метода Рунге-Кутты желательно описать и использовать одномерные вещественные массивы K1(2), K2(2), K3(2), K4(2) (или двумерный K(4,2)) и одномерный Y1(2). Последний применяется для передачи текущих значений в функцию F(X, Y). В подпрограммах по формулам соответствующих методов вычисляются новые значения Y(2) (y_1, y_2).

Головная программа должна содержать:

- описание используемых массивов (например, YE(2), YEK(2), YRK(2));
- ввод исходных данных X0, XK, M, Y0, P0;
- вычисление H и подготовку начальных значений для каждого метода (YE(1)=Y0, YE(2)=P0,...);
- в цикле по i от 1 до m вычисление решения уравнения методами Эйлера, Эйлера-Коши, Рунге-Кутта, точного решения, изменение X ($X=X+H$) и печать результатов.

В простейшем случае при оформлении процедур и головной программы можно совсем отказаться от использования массивов.

ВАРИАНТЫ ЗАДАНИЙ.

(во всех вариантах $X_0=0$.)

n	Диф.уравнение	Y0	P0	XK	Точное решение
1	$y''+y=1/\cos(x)$	1	0	0.5	$\cos(x)+x*\sin(x)+\cos(x)*\ln(\cos(x))$
2	$y''+2*y'+2*y=2*e^{-x}*\cos(x)$	1	0	0.5	$\exp(-x)*(\cos(x)+\sin(x)+x*\sin(x))$
3	$y''+4*y=\exp(3*x)*(13*x-7)$	0	-4	0.2	$\cos(2*x)-\sin(2*x)+\exp(3*x)*(x-1)$
4	$y''+4*y'+4*y=0$	1	-1	1	$(1+x)*\exp(-2*x)$
5	$y''-y=\sin(x)+\cos(2*x)$	1.8	-0.5	2	$2*\operatorname{ch}(x)-\sin(x)/2-\cos(2*x)/5$
6	$y''-3*y'=\exp(5*x)$	2.2	0.8	0.2	$2+0.1*(\exp(3*x)+\exp(5*x))$
7	$y''+4*y=\cos(3*x)$	0.8	2	1	$\cos(2*x)+\sin(2*x)-0.2*\cos(3*x)$
8	$y''-6*y'-y'=2*\exp(4*x)$	1.433	-3.67	1	$0.1*\exp(3*x)+\exp(-2*x)+\exp(4*x)/3$
9	$y''-2*y'+y=5*x*\exp(x)$	1	2	1	$\exp(x)+x*\exp(x)+5*\exp(x)*x^3/6$
10	$y''+y'-6*y=3*x^2-x-1$	-0.9	3.2	1	$0.1*\exp(2*x)-\exp(-3*x)-0.5*x^2$
11	$8*y''+2*y'-3*y=x+5$	1/9	-7/12	1	$\exp(x/2)+\exp(-3*x/4)-x/3-17/9$
12	$y''-5*y'+6*y=\exp(x)$	0	0	0.2	$-\exp(2*x)+\exp(3*x)/2+\exp(x)/2$
13	$y''-3*y'+2*y=x^2+3*x$	5.1	4.2	1	$\exp(x)+0.1*\exp(2*x)+x^2/2+3*x+4$
14	$y''+y=1+\exp(x)$	2.5	1.5	1	$\cos(x)+\sin(x)+1+\exp(x)/2$
15	$y''-2*y'+y=x*\exp(x)$	1	2	0.5	$(1+x)*\exp(x)+x^3*\exp(x)/6$
16	$y''-3*y'+2*y=2*\sin(x)$	2.6	3.2	1	$\exp(x)+\exp(2*x)+.2*\sin(x)+.6*\cos(x)$
17	$y''-y=\exp(2*x)*(x-1)$	11/9	-11/9	1	$2*\operatorname{ch}(x)+\exp(2*x)*(x/3-7/9)$
18	$y''+4*y'+4*y=2*x-3$	-1/4	-1/2	0.5	$(1+x)*\exp(-2*x)+x/2-5/4$

10. РЕШЕНИЕ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ.

Для решения систем нелинейных уравнений применяются итерационные методы: простых итераций, Ньютона и др.

В методе ПРОСТЫХ ИТЕРАЦИЙ система нелинейных уравнений (n - порядок системы)

$$f_1(x_1, x_2, \dots, x_n) = 0$$

$$f_2(x_1, x_2, \dots, x_n) = 0$$

.....

$$f_n(x_1, x_2, \dots, x_n) = 0$$

или в векторной форме

$$F(X) = 0$$

приводится к виду

$$x_1 = p_1(x_1, x_2, \dots, x_n)$$

$$x_2 = p_2(x_1, x_2, \dots, x_n)$$

.....

$$x_n = p_n(x_1, x_2, \dots, x_n)$$

или

$$X = P(X).$$

Например,

$$\sin(x_2 + 1) - x_1 = 1 \Rightarrow x_1 = \sin(x_2 + 1) - 1$$

$$2 * x_2 + \cos(x_1) = 2 \Rightarrow x_2 = (2 - \cos(x_1)) / 2.$$

Итерационный процесс $X^{i+1} = P(X^i)$ ведется по следующему алгоритму:

1. Задается начальное приближение $X = (x_1, x_2, \dots, x_n)$.
2. Вычисляется последующее приближение $Z = P(X)$ ($z_k = p_k(x_1, x_2, \dots, x_n)$, $k=1, \dots, n$).
3. Определяется величина расхождения E последующего приближения с предыдущим:
 $E = \sqrt{(\sum (z_k - x_k)^2) / n}$ или $E = \max(|z_k - x_k|)$.
4. В качестве начального приближения X для следующего шага итераций принимается значение Z : $x_k = z_k$.
5. Проверяется необходимость продолжения итерационного цикла: если $E > EPS$ (EPS - заданная точность вычислений), то цикл повторяется с пункта 2, в противном случае решение считается найденным. Так как итерационный процесс может расходиться, то в алгоритме необходимо предусмотреть выход из цикла до достижения заданной точности при

превышении максимально допустимого количества итераций MMAХ (см. задачу 8 - решение СЛАУ методом простых итераций).

При оформлении алгоритма в виде подпрограммы (см. задачу 8) среди входных параметров должно присутствовать имя подпрограммы, вычисляющей для заданного массива X массив значений P.

МЕТОД НЬЮТОНА решения систем нелинейных уравнений заключается в следующем. Пусть задано некоторое начальное приближение решения X. Уточнение решения ΔX можно получить, решая систему

$$F(X + \Delta X) = 0.$$

Раскладывая F в ряд Тейлора

$$F(X + \Delta X) = F(X) + \left[\partial F / \partial X \right] * \Delta X + \dots$$

и учитывая, что в левой части последнего равенства стоит 0, получим систему линейных алгебраических уравнений для определения ΔX

$$\left[\partial F / \partial X \right] * \Delta X = -F. \quad (1)$$

Элементы матрицы $\left[\partial F / \partial X \right]$ вычисляются по формулам

$$\left[\partial F / \partial X \right]_{ij} = \partial f_i / \partial x_j.$$

Уточненное значение решения Z после решения СЛАУ определяется как

$$Z = X + \Delta X. \quad (2)$$

Формулы (1), (2) можно объединить. Тогда для Z получим

$$Z = X - \left[\partial F / \partial X \right]^{-1} * F. \quad (3)$$

В качестве начального приближения для следующего шага итераций выбирается значение Z. Итерации прекращаются при достижении заданной точности ($\Delta x_i < EPS$). В подпрограмме метода Ньютона необходимо предусмотреть ограничение на максимальное количество итераций. Среди параметров процедуры указываются имена подпрограмм, вычисляющих значения одномерного массива F и двумерного $\left[\partial F / \partial X \right]$.

ЗАДАНИЕ. Решить две системы нелинейных уравнений (см. варианты заданий): первую - методом простых итераций; вторую - методом Ньютона по формулам (1),(2) с использованием п/п SIMQ, затем по формуле (3) с использованием п/п MINV. Для этого составить: подпрограммы вычисления F, P, $\left[\partial F / \partial X \right]$; подпрограммы методов простых итераций и Ньютона; головную программу. Провести счет. Начальное приближение для первой системы выбрать графически.

ВАРИАНТЫ ЗАДАНИЯ.

($x = x_1$, $y = x_2$)

№	система 1	система 2	№	система 2	система 2
1	$\sin(x+1)-y=1.2$ $2x+\cos y=2$	$\operatorname{tg}(xy+.45)=x^2$ $0.6x^2+2y^2=1$	11	$\sin(y+1)-x=1.2$ $y+\cos x-2=0$	$\operatorname{tg}(xy+0.2)=x^2$ $0.6 \cdot x^2+y^2=1$
2	$\cos(x-1)+y=0.5$ $x-\cos y=3$	$\sin(x+y)-1.6x=0$ $x^2+y^2=1$	12	$\cos(y-1)+x=0.5$ $y-\cos x=3$	$\sin(x+y)-1.5x=-0.1$ $x^2+y^2=1$
3	$\sin x+2y=2$ $\cos(y-1)+x=0.7$	$\operatorname{tg}(xy+0.1)=x^2$ $x^2+2y^2=1$	13	$\sin y+2x=2$ $\cos(x-1)+y=0.7$	$\operatorname{tg}(xy+0.4)=x^2$ $0.8x^2+2y^2=1$
4	$\cos x+y=1.5$ $2x-\sin(y-0.5)=1$	$\sin(x+y)-1.2x=0.2$ $x^2+y^2=1$	14	$\cos y+x=1.5$ $2y-\sin(x-0.5)=1$	$\sin(x+y)=1.2x-0.1$ $x^2+y^2=1$
5	$\sin(x+0.5)-y=1$ $\cos(y-2)+x=0$	$\operatorname{tg}(xy+0.3)=x^2$ $0.9x^2+2y^2=1$	15	$\sin(y+0.5)-x=1$ $\cos(x-2)+y=0$	$\operatorname{tg}(xy+0.1)=x^2$ $0.9x^2+2y^2=1$
6	$\cos(x+0.5)+y=0.8$ $\sin y-2x=1.6$	$\sin(x+y)=1.3x$ $x^2+y^2=1$	16	$\cos(y+0.5)+x=0.8$ $\sin x-2y=1.5$	$\sin(x+y)-1.4x=0$ $x^2+y^2=1$
7	$\sin(x-1)+y=1.3$ $x-\sin(y+1)=0.8$	$\operatorname{tg} xy=x^2$ $0.8x^2+2y^2=1$	17	$\sin(y-1)+x=1.5$ $y-\sin(x+1)=0.8$	$\operatorname{tg}(xy+0.1)=x^2$ $0.5x^2+2y^2=1$
8	$2y-\cos(x+1)=0$ $x+\sin y=-0.1$	$\sin(x+y)-1.5x+0.1$ $x^2+y^2=1$	18	$2x-\cos(y+1)=0$ $y+\sin x+0.4=0$	$\sin(x+y)=1.1x-0.1$ $x^2+y^2=1$
9	$\cos(x+0.2)-y=2$ $\sin(y)-2x-1=0$	$\operatorname{tg} xy=x^2$ $0.7x^2+2y^2=1$	19	$\cos(y+0.5)-x=2$ $\sin x-2y=1$	$\operatorname{tg}(x-y)-xy=0$ $x^2+2y^2=1$
10	$\sin(x+2)-y=1.5$ $x+\cos(y-2)=0.8$	$\sin(x+y)-1.2x=-.1$ $x^2+y^2=1$	20	$\sin(y+2)-x=1.5$ $y+\cos(x-2)=0.5$	$\sin(x-y)-xy=0$ $x^2+y^2=0.75$

11. РАЗЛОЖЕНИЕ ФУНКЦИЙ В РЯД ФУРЬЕ.

Периодическая функция $f(x)$ с периодом T может быть приближенно представлена в виде конечной суммы $p(x)$, получаемой при обрывании ряда Фурье на n -м члене

$$f(x) \cong p(x) = A_0/2 + \sum_{k=1}^n [A_k \cos(2\pi k x/T) + B_k \sin(2\pi k x/T)]$$

где

$$A_k = 2/T \int_0^T f(x, k, T) dx, \quad f_c(x, k, T) = f(x) \cos(2\pi k x/T), \quad k=0, 1, \dots, n$$

$$B_k = 2/T \int_0^T f_s(x, k, T) dx, \quad f_s(x, k, T) = f(x) \sin(2\pi k x/T), \quad k=1, 2, \dots, n$$

Степень точности приближения зависит от n . Погрешность при замене $f(x)$ на $p(x)$ можно оценить через среднее квадратичное отклонение:

$$\Delta = \sqrt{\frac{1}{T} \int_0^T [f(x) - p(x)]^2 dx}$$

ЗАДАНИЕ. Для своего варианта исходных данных:

- 1) Составить подпрограммы-функции для вычисления значений $f(x)$, $f_c(x, k, T)$, $f_s(x, k, T)$.
- 2) Составить подпрограмму для вычисления коэффициентов разложения функции в ряд Фурье.
- 3) Составить подпрограмму для вычисления значений функции $p(x)$ в зависимости от n .
- 4) Составить главную программу и вычислить на интервале $[0, T]$ значения функций $f(x)$ и $p(x)$ и среднее квадратичное отклонение для $n=1, 3, 20$.
- 5) По полученным результатам построить графики $f(x)$ и $p(x)$ для $n=1, 3, 20$.

ПРИМЕЧАНИЕ. Во всех вариантах принять $T=1$ и $\max|f(x)|=1$.

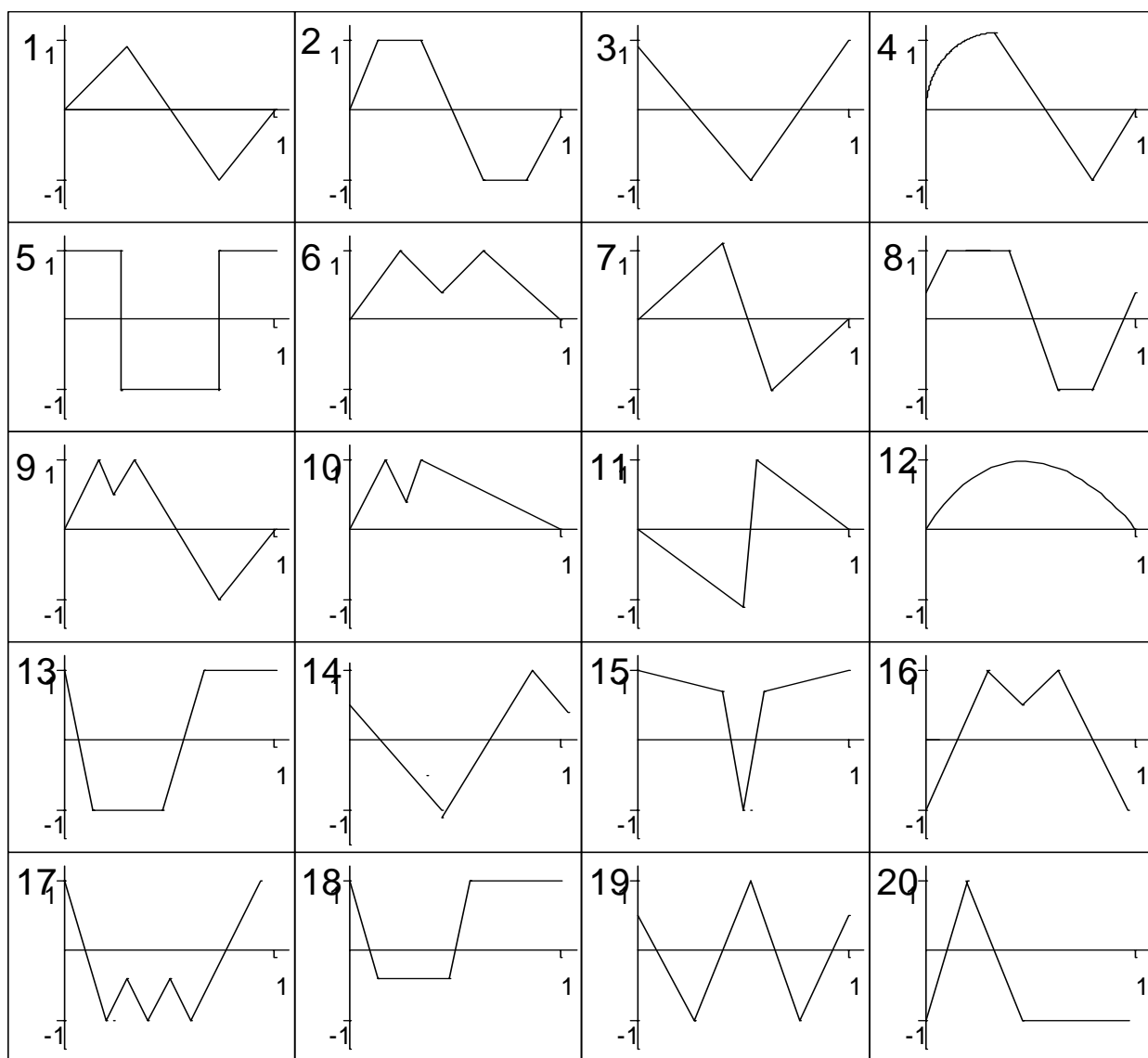
УКАЗАНИЯ к выполнению задания.

В качестве формального параметра подпрограмм-функций f , f_c , f_s используется только переменная x . Параметры k и T в функции f_c , f_s передаются через общую область. При вычислении коэффициентов разложения функции интегрирование проводить численно, желательно использовать стандартную программу метода Гаусса. В большинстве вариантов $f(x)$ задана аналитическими выражениями по нескольким участкам, поэтому интеграл от 0 до T необходимо вычислять как сумму интегралов по этим участкам. Так как подынтегральная функция является быстро осциллирующей (на отрезке $[0, T]$ имеет $2k$ полувольт), для

увеличения точности интегрирования каждый участок необходимо разбить, как минимум, на k промежутков и на каждом применять процедуру численного интегрирования.

В головной программе при вычислении значений функций $f(x)$ и $p(x)$ шаг изменения параметра x необходимо принять не более 0.05 (при графическом отображении результатов на экране шаг изменения x принять равным 0.02). Печать результатов в файл предусмотреть в форме таблицы. В этой же таблице распечатать значения коэффициентов A_k и B_k . Интеграл при определении среднего квадратичного уклонения можно вычислить по правилу трапеций, используя для этого найденные значения $f(x)$ и $p(x)$.

ВАРИАНТЫ ЗАДАНИЯ



12. ОПРЕДЕЛЕНИЕ ЭКСТРЕМУМОВ ФУНКЦИИ МЕТОДОМ ЗОЛОТОГО СЕЧЕНИЯ.

ЗАДАНИЕ. На отрезке $[a,b]$ найти: а) все экстремумы функции $f(x)$; б) максимальное значение модуля функции $G=\max|f(x)|$. Построить график функции $f(x)/G$.

Алгоритм поиска экстремумов состоит из двух этапов:

I) Выделение (локализация) внутри $[a,b]$ таких отрезков $[c,d]$, каждый из которых содержит только один экстремум. Это можно сделать, отделяя корни уравнения $f'(x)=0$ (см. тему "Решение нелинейных уравнений"). При этом производную можно заменить конечной разностью $f'(x)=[f(x+t)-f(x)]/t$, где t - малая величина (например, $t=0.0001$).

При составлении подпрограммы отделения точек экстремума в качестве параметров подпрограммы желательно указать следующие:

Входные параметры:

$a1, b1$ - граничные точки отрезка, внутри которого ищутся экстремумы;

f - имя функции;

h - шаг перебора значений x при анализе поведения функции $f(x)$.

Выходные параметры:

c, d - границы отрезка внутри $[a1,b1]$, содержащий первый (справа от $a1$) экстремум;

m - переменная, по значению которой можно судить, содержит ли отрезок $[c,d]$ \max или \min : $m=1$ соответствует \max , а $m=-1$ - \min ;

ier - код ошибки; $ier=0$ если $[c,d]$ найден; $ier=1$ если экстремума на $[a1,b1]$ нет.

В качестве алгоритма отделения экстремума можно предложить следующий: задать $x=a1$ и вычислить $y=[f(x+t)-f(x)]/t$; менять x с шагом h $x=x+h$ до тех пор, пока либо произведение $y*[f(x+t)-f(x)]/t$ не станет меньше 0, либо x не выйдет за пределы отрезка $[a1,b1]$, т.е. x не станет больше $b1$; в первом случае $c=x-h, d=x, ier=0$; если значение $[f(c+t)-f(c)]/t$ больше 0, то $m=1$, иначе $m=-1$; во втором случае $ier=1$ и осуществляется возврат в вызывающую программу.

II) Уточнение до заданной точности экстремума функции $f(x)$ методом золотого сечения.

Золотым сечением отрезка называют деление отрезка на две части так, что отношение длины всего отрезка к длине большей его части равно отношению длины большей части к меньшей. Золотое сечение отрезка $[c,d]$ производят две симметрично расположенные точки $x1, x2$:

$x_1=c+(1-v)*(d-c)$, $x_2=c+v*(d-c)$, где $v=(-1+\sqrt{5})/2=0.6180339$. При этом x_1 производит золотое сечение $[c,x_2]$, а x_2 - золотое сечение $[x_1,d]$.

При составлении подпрограммы метода золотого сечения в качестве параметров подпрограммы желательно указать следующие:

Входные параметры:

c, d - граничные точки отрезка, внутри которого содержится только один экстремум;

f - имя функции;

ϵ - точность вычисления абсциссы экстремума;

Выходные параметры:

x_0 - абсцисса точки экстремума;

m - переменная, по значению которой можно судить, является ли найденный экстремум \max или \min : $m=1$ соответствует \max , а $m=-1$ - \min ;

ier - код ошибки; $ier=0$ если экстремум найден; $ier=1$ если экстремума на $[c,d]$ нет.

Опишем возможный алгоритм поиска \max функции $m*f(x)$ на участке $[c,d]$ ($m=\{1,-1\}$). а) выполним проверку наличия экстремума на $[c,d]$: если $[f(c+t)-f(c)] * [f(d+t)-f(d)]>0$, то $ier=1$ и осуществляется возврат в вызывающую программу; в противном случае $ier=0$ и осуществляется поиск экстремума;

б) для сохранения исходных значений c, d определим новые переменные: $c_1=c, d_1=d$.

с) вычислим значение m : если $f(c_1+t)-f(c_1)>0$, то $m=1$, иначе $m=-1$.

д) вычислим значения x_1, x_2 : $x_1=c_1+(1-v)*(d_1-c_1)$, $x_2=c_1+v*(d_1-c_1)$.

е) вычислим значения функции в точках x_1, x_2 : $y=m*f(x_1)$, $z=m*f(x_2)$.

ф) если $y<z$ то: $c_1=x_1$, $x_1=x_2$, $y=z$, $x_2=c_1+v*(d_1-c_1)$, $z=m*f(x_2)$;

иначе: $d_1=x_2$, $x_2=x_1$, $z=y$, $x_1=c_1+(1-v)*(d_1-c_1)$, $y=m*f(x_1)$.

г) если $d_1-c_1>\epsilon$ повторяем (ф).

х) за точку \max функции $m*f(x)$ принимаем точку $x_0=(d_1+c_1)/2$.

При вычислении $G=\max|f(x)|$ надо иметь в виду, что функция $f(x)$ может достигать \max (\min) либо в точках экстремумов, либо на концах отрезка $[a,b]$.

При выполнении задания НЕОБХОДИМО составить: а) подпрограмму локализации экстремумов функции $f(x)$; б) подпрограмму метода золотого сечения; в) подпрограмму-функцию $f(x)$; г) головную программу для определения экстремумов функции $f(x)$, максимального значения модуля функции $G=\max|f(x)|$, вычисления на отрезке $[a,b]$ (табулирования) значений функции $f(x)/G$.

В головной программе при первом обращении к подпрограмме отделения экстремума в качестве отрезка $[a_1, b_1]$ необходимо выбрать заданный отрезок $[a, d]$ ($a_1=a, b_1=b$). При повторном обращении к этой же подпрограмме переменной a_1 присваивается значение переменной d , вычисленной на предыдущем шаге поиска экстремумов ($a_1=d$). Поиск экстремумов прекращается, если $ier=1$.

ВАРИАНТЫ ЗАДАНИЯ.

Для всех вариантов $a=0.$, $b=1$.

Функция $f(x)$ задается в виде:
$$f(x) = \sum_{i=1}^3 u_i * \sin[\pi * (v_i * x + w_i)].$$

Величины u_i , v_i , w_i для $i=1,2,3$ вычисляются по формулам:

$$u_i = 1 + (0.3 - p_1) * (i - 1), \quad v_i = 2 * i - (i - p_2) / 2 + (i + p_4) / 3, \quad w_i = i / 6 - (i - p_3) / 3 + (i + p_5) / 2$$

Значения p_1 :- p_5 определите следующим образом: переведите свой номер по журналу в двоичный код, допишите перед полученным числом нули так, чтобы у Вас получился пятизначный код из нулей и единиц. Значение p_1 будет равно цифре в первой позиции кода, p_2 - во второй позиции, и т.д. Например, если Ваш номер по журналу - 9, то Ваш код будет 01001, значения p равны: $p_1=0$, $p_2=1$, $p_3=0$, $p_4=0$, $p_5=1$, а функция $f(x)$ примет вид $f(x) = \sin[\pi * (14 * x + 5) / 6] + 1.3 * \sin[\pi * (25 * x + 7) / 6] + 1.6 * \sin[\pi * (36 * x + 9) / 6]$, $\pi = 3.1415926$.

13. ОПРЕДЕЛЕНИЕ СОБСТВЕННЫХ ЧАСТОТ И ФОРМ КОЛЕБАНИЙ ВАЛА.

Рассматривается задача определения собственных частот и форм колебаний вала, концы которого упруго закреплены с помощью жесткостей K_1 и K_2 . На концах вала насажены жесткие диски с моментами инерции J_1 и J_2 . Уравнение для определения собственных форм крутильных колебаний вала имеет вид

$$y''(x) + a^2 y(x) = 0 \quad (1)$$

где x - безразмерная координата с пределами изменения от 0 до 1;

$y(x)$ - разыскиваемая функция - амплитуда угла поворота сечения вала в точке x ;

a - параметр, подлежащий определению - безразмерная частота ($a > 0$).

Решение уравнения (1) записывается как:

$$y(x) = C_1 \sin(ax) + C_2 \cos(ax) \quad (2)$$

Неизвестные постоянные C_1 , C_2 определяются из граничных условий:

$$y'(x) = -a^2 b_1 y(x) + g_1 y(x), \quad \text{при } x=0 \quad (3)$$

$$y'(x) = a^2 b_2 y(x) - g_2 y(x), \quad \text{при } x=1 \quad (4)$$

Безразмерные параметры b_1 , b_2 определяются как отношения моментов инерции дисков к приведенному моменту инерции вала, а g_1 , g_2 - как отношения жесткостей K_1 , K_2 к жесткости вала (величины b_1 , b_2 , g_1 , g_2 задаются для каждого варианта задания).

Подставляя (2) в условия (3), (4), получим систему однородных линейных алгебраических уравнений относительно C_1 , C_2 , которую в общем случае можно записать в виде

$$d_{11}(a) C_1 + d_{12}(a) C_2 = 0 \quad (5)$$

$$d_{21}(a) C_1 + d_{22}(a) C_2 = 0$$

Система (5) имеет ненулевое решение, когда определитель, составленный из коэффициентов при C_1 , C_2 , равен нулю. Раскрыв определитель, получим нелинейное уравнение относительно параметра a

$$f(a) = d_{11}(a) d_{22}(a) - d_{12}(a) d_{21}(a) = 0 \quad (6)$$

Полученное нелинейное уравнение имеет бесконечное множество корней a_i , $i=1, 2, \dots$, которые можно отделить и уточнить до требуемой точности любым известным методом. Для каждого a_i из какого-либо уравнения системы (5) можно определить отношение C_2/C_1 (или C_2/C_1). Например, из первого уравнения (5) получим $C_2/C_1 = -d_{11}(a_i)/d_{12}(a_i)$. С учетом последнего условия уравнение (2) переписывается как

$$y(x) = C_1 \left[\sin(a_i x) - d_{11}(a_i)/d_{12}(a_i) \cos(a_i x) \right] \quad (7)$$

Собственные формы (7) колебаний вала определяются с точностью до множителя C_1 . В вариантах задания предлагается вычислить C_1 , используя один из двух способов нормирования:

$$\text{либо } \max|y(x)|=1, \quad \text{тогда } C_1=1/\max|w(x)| \quad (8)$$

$$\text{либо } \int_0^1 y(x)^2 dx = 1, \quad \text{тогда } C_1 = \sqrt{1 / \int_0^1 w(x)^2 dx} \quad (9)$$

$$\text{где } w(x) = \sin(a_i * x) - d_{11}(a_i)/d_{12}(a_i) * \cos(a_i * x) \quad (10)$$

Определить максимальное значение модуля функции можно, вычислив все модули экстремумов функции, затем из них и модулей значений функции на концах отрезка $[0,1]$ найти максимальный. Экстремумы можно найти, используя метод золотого сечения (см.задачу 12), либо вычисляя корни производной функции (см.задачу 5). В простейшем случае $\max|w(x)|$ определяется простым перебором значений функции при изменении x с достаточно малым шагом аналогично тому, как это делается при поиске максимального элемента массива (см.задачу 4).

Собственная форма $w(x)$ может иметь узлы (точки, в которых $w(x)$ меняет знак), поэтому при использовании формулы (9) интеграл от 0 до 1 необходимо вычислять как сумму интегралов по участкам между узлами. Для определения узлов (корней) $w(x)$ используется метод половинного деления, Ньютона и т.д. (см.задачу 5). В простейшем случае C_1 можно найти, вычисляя интеграл от 0 до 1 методом Симпсона с достаточно малым шагом (см.задачу 6).

ЗАДАНИЕ. Для своего варианта исходных данных:

- 1.Записать граничные условия в виде (3),(4).
- 2.Записать систему однородных линейных алгебраических уравнений в виде (5).
- 3.Получить нелинейное уравнение в виде (6).
- 4.Записать функцию $w(x)$ в виде (10).
- 5.Составить программу для: определения трех первых положительных корней a_i уравнения (6); определения по формулам (8) либо (9) коэффициентов C_{1i} для каждого a_i ; вычисления значений (табулирования) функции (7) при изменении x от 0 до 1 для найденных a_i , C_{1i} (результаты выводить в форме таблицы).
- 6.Провести счет. Результаты счета оформить графически.

При составлении программы отдельные алгоритмы (отделения корней, уточнения корней и др.) оформить в виде подпрограмм.

ВЫБОР варианта задания. Запишите Ваш номер по журналу в двоичном коде. Запишите нули перед кодом до пяти позиций. Численные значения в каждой позиции кода определяют (слева направо): значения b_1, g_1, g_2, b_2 , способ нормирования (если в последней позиции стоит 1, то C_1 вычисляется по формуле (8), если 0 - по формуле (9)).

ПРИМЕР. Для варианта 10 получим код 01010. Значения параметров будут следующие: $b_1=0, g_1=1, g_2=0, b_2=1$. Для вычисления C_1 используется формула (9).

Из условия (3) получим: $C_1*a=C_2$ или $C_1*a-C_2=0$, т.е. $d_{11}=a, d_{12}=-1$.

Из условия (4) получим: $C_1*(\cos(a)-a*\sin(a)) + C_2*(-\sin(a)-a*\cos(a))=0$,
 $d_{21}=\cos(a)-a*\sin(a), d_{22}=-\sin(a)-a*\cos(a)$.

Нелинейное уравнение (6) запишется в виде: $(1-a^2)*\cos(a)-2*a*\sin(a)=0$.

Для каждого корня a_i последнего уравнения C_{1i} определяется по формуле:

$$C_{1i} = \sqrt{1 / \int_0^1 [\sin(a_i * x) + a_i * \cos(a_i * x)]^2 dx}$$

Собственные функции для каждой пары значений a_i, C_{1i} вычисляются по формуле:

$$y(x) = C_{1i} * [\sin(a_i * x) + a_i * \cos(a_i * x)].$$

Для выполнения расчетов необходимо составить:

- подпрограммы-функции для вычисления значений функций $f(a), w(x)$ (или $w(x)^2$);
- подпрограммы отделения корней нелинейного уравнения, уточнения корней методом половинного деления, определения максимального значения модуля функции (или численного интегрирования);
- главную программу.

Главная программа должна содержать описание:

- массивов $A(3), C_1(3)$ для хранения значений a_i, C_{1i} , а также описание массива $Y(3)$, используемого при печати результатов в форме таблицы;
- внешних функций, используемых при решении нелинейного уравнения и при поиске максимального модуля функции (или при численном интегрировании);
- общей области, используемой для передачи текущего значения a_i в подпрограммы-функции;
- оператора-функции для вычисления $y(x)$.

В теле главной программы вычисляются:

- в цикле по параметру i от 1 до 3 значения a_i, C_{1i} ; результаты (в том числе и промежуточные
- точки экстремумов, узлы формы - если они вычислялись) записываются в выводной файл;

- в цикле по параметру x от 0 до 1 с шагом 0.05 значения трех функций $y(x)$ для каждой пары a_i , $C1_i$ (цикл по $i=1,3$ - внутренний по отношению к циклу по x); результаты выводятся в файл в форме таблицы.