МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«_4_» _03_____ 2019 г.

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Методические рекомендации по самостоятельной работе бакалавров направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» по дисциплине «Вычислительная математика»

Курск 2019

УДК 519.6

Составитель: Е.П. Кочура

Рецензент
Кандидат технических наук, доцент *И.Н. Ефремова*

Вычислительная математика: методические рекомендации по самостоятельной работе бакалавров направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» по дисциплине «Вычислительная математика» / Юго-Зап. гос. ун-т; сост.: Е.П. Кочура. – Курск, 2019. – 11 с.

Изложены основные требования к организации самостоятельной работы студентов. Перечислены виды и формы проведения самостоятельной работы и ее контроля, раскрыты особенности организационно-методического обеспечения, дается краткое описание алгоритмического языка Паскаль, темы самостоятельной работы и библиографический список.

Материал предназначен для бакалавров направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия».

Текст печатается в авторской редакции

Подписано в печать 4.03.19. Формат 60 х 84 1/16.
Усл. печ. л. 1,2. Уч.- изд. л. 1,1. Тираж 50 экз. Заказ 143
Бесплатно.
Юго-Западный государственный университет.
305040, Курск, ул. 50 лет Октября, 94.

ВВЕДЕНИЕ

В настоящее время трудно найти такую область человеческой деятельности, где не используется вычислительная техника. Поэтому для современного человека умение использовать ЭВМ для стоящих перед ним задач является необходимым. Вычислительная техника используется в производстве, науке и технике. Для решения таких задач строятся их математические модели и решаются с применением методов вычислительной математики.

«Вычислительная математика» представляет собой дисциплину вариативной части учебного плана и изучается бакалаврами направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия».

1.1. Цель дисциплины

Развитие логического и алгоритмического мышления, овладение основными вычислительными методами математики и их реализации на ЭВМ.

1.2. Задачи дисциплины

-овладение методами математического анализа и математического моделирования;

-освоение студентами методики применения программных средств для решения научных и практических задач;

-выработка у студентов умения самостоятельно расширять математические знания, проводить математический анализ и решать на ЭВМ прикладные (инженерные) задачи.

1.3. Планируемые результаты обучения

Знать:

-источники и квалификацию погрешностей математического моделирования,

-особенности математических вычислений реализуемых на ЭВМ,

-основные понятия и методы аппроксимации,

-методы численного дифференцирования и интегрирования,

-методы решения систем линейных уравнений и нелинейных уравнений,

-методы одномерной и многомерной оптимизации,

-методы решения обыкновенных дифференциальных уравнений.

Уметь:

-использовать типовые и широко распространенные программные продукты в научной и практической деятельности,

-оценивать точность численных результатов математического моделирования,

-применять математические методы и вычислительную технику для решения практических задач.

Владеть:

-методами решения и исследования с помощью ЭВМ прикладных задач по специальности,

-вычислительными методами при решении теоретических и практических задач;

-навыками реализации вычислительных методов на ЭВМ.

У обучающихся формируются следующие компетенции:

- способность осваивать методики использования программных средств для решения практических задач (ОПК-2);

- способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности (ПК-3).

ОБЩИЕ ПОЛОЖЕНИЯ

Самостоятельная работа студентов (СРС) - одна из форм индивидуальной работы, важнейшая составная часть процесса подготовки будущих специалистов.

Целями СРС являются формирование у студентов навыков к самостоятельному творческому труду, умение решать профессиональные задачи с использованием всего арсенала современных средств, потребность к непрерывному самообразованию и совершенствованию своих знаний; приобретение опыта планирования и организации рабочего времени и расширение кругозора.

Самостоятельная работа студентов способствует активизации умственной деятельности и самостоятельному усвоению знаний, формированию профессиональных умений и навыков, обеспечивает формирование общекультурных, профессиональных компетенции будущего специалиста. Она максимально развивает познавательные и творческие способности личности в рамках актуализации компетентностного подхода.

Кроме того, СРС позволяет студенту развивать свои возможности, потребности, интересы посредством проектирования собственного индивидуального образовательного маршрута, побуждает к научно-исследовательской работе.

Самостоятельная работа студентов включает в себя два вида: аудиторную и внеаудиторную работу.

Самостоятельная аудиторная работа студентов (САРС) по дисциплине выполняется под непосредственным руководством и контролем преподавателя, по его заданию. САРС осуществляется в сроки, определяемые учебным планом и расписанием занятий.

Внеаудиторная самостоятельная работа выполняется студентами по заданию преподавателя, но без его непосредственного участия и не регламентируется расписанием занятий. Она может выполняться студентами с использованием дистанционных образовательных технологий в различных формах, главным принципом которых является удаленная СРС, где студент и преподаватель взаимодействуют (передают и получают задания, методические материалы, контрольные вопросы, тестовые задания и т. п. в электроном виде) посредством локальной и глобальной сетей. Формами реализации такой работы могут быть различные способы IT-коммуникаций, выбираемые преподавателем с учетом особенностей преподавания дисциплины.

Объем времени на САРС включается в общий объем времени, отведенного на СРС, согласно учебному плану. При этом на САРС не переносятся лабораторные, практические, семинарские и другие занятия, предусмотренные расписанием.

Самостоятельная аудиторная работа студентов включает следующие формы работ:

- дополнительные занятия;
- текущие консультации по дисциплине;
- консультация и прием индивидуальных домашних заданий.

Внеаудиторная СРС, в том числе с использованием дистанционных образовательных технологий, включает следующие формы работ:

- работа с учебниками, учебными и методическими пособиями (как на бумажных, так и на электронных носителях);
- работа с первоисточниками;
- работа с конспектами лекций, научными статьями;
- составление конспектов в виде электронного документа, презентаций на базе рекомендованной лектором учебной литературы, включая электронные учебные издания (электронные учебники, курсы, презентации, модели, анимированные изображения, видео - кейсы, библиотеки, контрольно-измерительные материалы и др.);
- подготовка к лабораторным занятиям, в том числе по материалам электронных учебных изданий, специализированных тематических сайтов, электронных копий научных статей и т. п.;
- составление отчетов по лабораторным работам;
- научный эксперимент, размышления и обсуждения, выполнение
- осуществление самоконтроля (компьютерное тестирование и т. д.);

  подготовка к тестированию;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, индивидуальных работ по отдельным разделам содержания дисциплин и т. д.;

- проработка тем, вынесенных в рабочей программе дисциплины на самостоятельное изучение.

Формы, объем и содержание заданий по СРС устанавливается кафедрой в соответствии с учебными планами и рабочими программами учебных дисциплин.

ПЛАНИРОВАНИЕ СРС

Основой для планирования СРС являются:

- федеральный государственный образовательный стандарт высшего профессионального образования (ФГОС ВПО) и государственный образовательный стандарт высшего профессионального образования (ГОС ВПО);

- учебный план специальности (направления подготовки);

- рабочая программа дисциплины.

В соответствии с требованиями ГОС ВПО и ФГОС ВПО объем изучаемых дисциплин в рабочих учебных планах установлен (нормирован) в академических часах и включает в себя аудиторную и самостоятельную (внеаудиторную) работу студентов. Трудоемкость самостоятельной работы по дисциплине определяется из рабочих учебных планов.

Затраты времени на выполнение всех форм СРС по каждой дисциплине строго соответствуют действующему учебному плану специальности (направления подготовки), а содержание - требованиям основной образовательной программы ВПО.

Методика планирования самостоятельной работы складывается из следующих элементов:

$$T_{СУМ}=T_{ЛП} + T_{СП} + T_{ЗЭ} + T_{ИЗ},$$

$T_{СУМ}$ – суммарное время на СРС по данной дисциплине, определенное учебным планом, ч;

$T_{ЛП}$ – время на подготовку к лекциям, лабораторным, практическим, семинарским занятиям, ч;

$T_{СП}$ – время на самостоятельное изучение разделов и тем учебной дисциплины;

$T_{ЗЭ}$ - время на подготовку к зачетам и экзаменам;

$Т_{ИЗ}$ - время на самостоятельное выполнение индивидуальных заданий (курсовой проект, курсовая работа, расчетно-графическая работа, конспект, реферат, упражнение и др.).

Сведения о СРС указываются в рабочей программе каждой дисциплины и утверждаются зав. кафедрой и деканом до начала учебного семестра. В них указываются перечень выполняемых работ, их содержание, объем заданий в часах, сроки выполнения и проведения контроля.

После ознакомления с этой информацией, каждый студент составляет график самостоятельной работы и график сдачи модулей с указанием сроков их выполнения.

При составлении графика СРС необходимо исходить из условий:

- согласования сроков выполнения СРС по всем дисциплинам;

- обеспечения ритмичности работы в течение семестра;

- отсутствия перегрузки заданиями в течение какой-либо недели.

Рекомендуется планировать завершение на одной неделе не более 2 заданий по СРС.

2. ОРГАНИЗАЦИОННО - МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СРС

Организационно-методическое обеспечение СРС включает разработку и проведение комплекса мероприятий по планированию и организации СРС:

- планирование СРС;

- обеспечение учебной литературой, методическими пособиями, в том числе электронными учебными изданиями, компьютерной техникой, программными продуктами;

- создание учебно-лабораторной базы и ее оснащение в соответствии с содержанием самостоятельной работы по курсам учебных дисциплин;

- создание необходимых условий для СРС в общежитиях, библиотеках, читальных залах, компьютерных классах.

Активизация СРС при проведении различных видов учебных занятий включает:

- переработку учебных планов и программ в рамках существующих ГОСов и ФГОСов с целью увеличения доли СРС. При этом должна учитываться обеспеченность тем и разделов учебной литературой и ее доступность для всех обучающихся;

- оптимизацию методов обучения, внедрение в учебный процесс современных образовательных и информационных технологий с учетом компетентностного подхода;

- разработку собственных электронных учебных изданий на основе имеющихся инструментов и средств;

- совершенствование системы текущего оперативного контроля СРС в течение семестра (использование возможностей балльно-рейтинговой системы, компьютеризированного тестирования и др.);

- совершенствование методики проведения практик и научно-исследовательской работы студентов;

- модернизацию системы курсового и дипломного проектирования для увеличения самостоятельности студентов на всех этапах работы.

Работа по учебно-методическому и техническому обеспечению СРС включает:

- определение тем дисциплины для самостоятельного изучения;

- определение форм самостоятельной работы;

- определение приемов контроля результатов СРС;

- техническое обеспечение СРС с использованием дистанционных образовательных технологий;

- обучение и консультация профессорско-преподавательского состава по разработке электронных учебных изданий и применению дистанционных образовательных технологий;

- разработка нового специализированного ПО.

Руководство СРС осуществляется преподавателями кафедры. В функции преподавателя входит:

- разработка календарно-тематического плана выполнения СРС по учебному курсу;

- определение объема учебного содержания и количества часов, отводимых на СРС, с учетом компетентностного подхода;

- подготовка пакета контрольно-измерительных материалов и определение периодичности контроля;

- определение системы индивидуальной работы со студентами.

Мониторинг СРС предусматривает организацию и корректировку учебной деятельности студентов, помощи при возникающих затруднениях. Контроль СРС предусматривает соотнесение содержания контроля с целями обучения; соответствие предъявляемых заданий тому, что предполагается проверить; дифференциацию контрольно-измерительных материалов.

К видам контроля СРС относятся

- текущий (оперативный) контроль;
- рубежный контроль;
- итоговый контроль (зачет, экзамен);
- самоконтроль.

Формами контроля СРС являются

- устный контроль;
- письменный контроль;
- тестовый контроль.

В качестве примеров можно привести индивидуальные собеседования, проверка выполнения домашних заданий, дискуссия, решение задач, защита лабораторных работ и др.

КРАТКОЕ ОПИСАНИЕ АЛГОРИТМИЧЕСКОГО ЯЗЫКА ПАСКАЛЬ

Общая структура программы на языке Паскаль

Программа представляет собой некоторый текст со своими правилами и грамматикой.

1. Первая строчка программы на языке Паскаль всегда начинается со служебного слова **program**. Далее после пробела следует имя (заголовок) программы и заканчивается первая строчка точкой с запятой.

Пример: Для программы с именем **gaus** первая строчка будет выглядеть следующим образом:

program gaus;

2. Обычно после имени программы следует **комментарий**, который предназначен для пояснения того, что делает программа, ее особенности и т.п. Комментарий не дает никаких указаний компьютеру, так как он не оказывает никакого влияния на работу программы. Наличие комментария создает определенные удобства как для программиста, так и пользователя программы. Комментарий помещается в **фигурные скобки { }**.

Пример:

{Программа решения линейной системы уравнений методом Гаусса**}**

3. Затем необходимо указать имена и типы всех величин, которые будут использованы в данной программе. Такое указание называется **объявлением** и начинается со служебного слова, которое зависит от характера объявления.

Для объявления переменных величин используется служебное слова **var**, после которого за пробелом следует список имен этих величин, разделенных запятой, потом между двумя пробелами стоит двоеточие, далее указывается тип переменных и объявление заканчивается точкой с запятой. В программе может быть много объявлений.

Пример:

var имена : тип**;**

Если в программе требуется записать или считать текстовую информацию из **внешнего файла**, то этому файлу в программе с помощью объявления также необходимо **присвоить** некоторое **имя** и указать **тип** имя.

В программе могут использоваться метки. Для объявления меток служит слово **label,** после которого следует имя метки.

Пример:
label ....;
имя

4. Начало исполняемой части программы (основная программа) начинается строчкой со служебным словом **begin.**
5. Далее следует **текст** основной программы.
6. После текста основной программы следует строчка со служебным словом **end** с точкой.

Служебные слова **begin** и **end** (без точки) могут использоваться в основной программе для выделения ее отдельных блоков.

Таким образом, общая структура программы имеет вид:

```
program  gaus;
{комментарий}
  var  .......;
  label  ....;
    begin

      ...
      ...      текст  основной  программы
      ...
    end.
```

Переменные. Переменная - это имя некоторой ячейки памяти или набора ячеек, в которых хранится значение этой переменной. Само название говорит о том, что значение переменной с этим именем может изменяться, т.е. будет изменяться содержание соответствующих ячеек памяти.

Имя переменной должно начинаться с буквы, далее могут следовать буквы или цифры от 0 до 9. Максимальная длина имени (количество букв и цифр в нем) не должно превышать 127.

Пример: x, y, wood, a123, ver23tt и т.п.

По типу своих значений переменные разделяются на следующие:

integer (целые числа);

real (вещественные числа);

boolean (булевы переменные), принимающие два значения: истина (1, да) - служебное слово **true** или ложь (0, нет) - служебное слово **false**;

text (для букв и символов)

и т.п.

По количеству значений, которые могут содержать переменные, они разделяются на простые и переменные с индексами (массивы). Каждая **простая переменная** может содержать только **одно значение**, а **переменная с индексами - много значений**.

При объявлении **простых** переменных за служебным словом **var** после пробела указывается только **имя каждой** переменной (имена разделяются запятыми), далее через пробел следует двоеточие, затем следует пробел и тип переменных, в конце точка с запятой.

Пример Объявление простых целых переменных с именами **x,good** и простых вещественных переменных с именами **z1,y,jk** в программе выглядит следующим образом:

var x,good **: integer;**

var z1,y,jk **: real;**

Для **переменных с индексом** за именем в квадратных скобках указывается **список индексов**. Значения индексов (целые числа) определяют одну конкретную ячейку в наборе из ячеек с этим именем. Индексы должны быть обязательно объявлены в начале программы как простые целые переменные.

При объявлении переменных с индексами после имени идет двоеточие между пробелами, далее служебное слово **array,** после которого в квадратных скобках для каждого индекса указывается два целых числа (эти числа указывают пределы, в которых может изменяться значение индекса), разделенных двоеточием или двумя точками. Далее, за пробелом, служебное слово **of,** а затем служебное слово, указывающее тип переменной. Если индексов больше одного, то такие пары чисел разделяются запятыми; при

этом порядок следования должен соответствовать порядку следования индексов.

Пример Объявление двух целых переменных с одним индексом x[j] и двумя индексами week[ii,j], а также двух вещественных переменных с тремя индексами z1[k1,m1,n1], z2[i,j,k] будет иметь вид:

var x **: array**[0:10] **of integer;**
var week **: array**[1:100,0:4] **of integer;**
var z1,z2 : **array**[3:5,0:4,1:10] **of real;**

Объявление переменных может быть произведено с использованием оператора **type**, за которым следует имя объявляемого типа знак равенства и тип.

Пример
type arr=**array** [0..100] **of real;**
В этом случае объявление переменных xi,u0,u1,an,bn,cn,dn данного типа в программе будет имеет вид.
var xi,u0,u1,an,bn,cn,dn**: arr;**

Операция присвоения. Переменные получают свои конкретные значения с помощью операции присвоения, которая обозначается символом ” **:=** “.

Пример:
y1**:=**-12;
x**:=**0.05; ,res[6,7,0]**:=**23.154; ,tt**:=**1;
z**:=**tt; (перед этой строчкой в программе переменной tt должно быть уже присвоено значение).

Метки. Для того чтобы в тексте программы пометить ту или иную выполняемую операцию (например, присвоения) служат метки. Имя метки начинается с буквы. Имена всех используемых меток объявляются в начале программы после служебного слова **label**. Приведем пример объявления меток с именами m1 и m2:
label m1,m2**;**

В **тексте** программы после **имени метки стоит двоеточие**, а далее следует помеченный оператор:

m1: x:=1**;**

m2: wed:=temp**;**

§4. Арифметические операции. Символы основных арифметических операций языка Паскаль приведены в следующей таблице:

Операции	Алгебраическое выражение	Паскаль	Примеры
Сложение	a+b	a+b	y+xr, mod+ref
Вычитание	a-b	a-b	fr-z, tr-ytx
Умножение	a·b, ab, a(b)	a∗b	xr∗mod, waz∗y
Деление	a:b, $\frac{a}{b}$	a/b	y/zz, ast/tooth

Пример. Написать текст блока программы, в котором производятся вычисления, по формуле

$$z = (x + y)\frac{(10.0 : y - 0{,}3x)}{(\sin x + y)}, \qquad x = 3, \qquad y = 0{,}1.$$

Возможны следующие варианты текстов для вычислений по этой формуле:

```
begin
x:= 3;
y:= 0.1;
z:= (x + y)*(10./y – 0.3* x) / (sin(x) + y);
end
```

или

```
begin
ar:= 3;
bt:= 0.1;
wood:= (ar + bt)*((10./bt – 0.3*ar) / (sin(ar) + bt));
end
```

В первом случае результат будет находиться в ячейке с именем z, а во втором случае - в ячейке с именем wood.

Операторы отношений. Ниже перечислены шесть основных операторов отношений и выражений с ними:

Отношение	Оператор Паскаля	Примеры выражений с операторами отношений (условные выражения)
Равно	=	x=3, wer=sqrt
Не равно	<>	x<>3, ten<>z1
Меньше	<	6 < x, a+b<z-5
Меньше или равно	<=	ar<=23, x+1<=y
Больше	>	ab>0, zz>v+0.1
Больше или равно	>=	x>=4, yr>=z-1

Выражение с оператором отношений (условное выражение) является булевой переменной. Если оно выполняется, то это выражение принимает значение **true** (истина), если не выполняется, то принимает значение **false** (ложь).

Логические или булевы операции. В Паскале рассматриваются три логические (булевы) операции с булевыми переменными: “и” (конъюнкция), “или” (дизъюнкция), и “не” (отрицание). Как мы уже говорили, булевой переменной может быть и условное выражение.

Логическая операция	Логическое выражение	Паскаль	Примеры
и	a “и” b	a **and** b	(x+1=y) and (z>1), xx and c
или	c “или” d	c **or** d	(we>=1) or (dd<>c1), zx or res
не	“не” y	**not** y	

			not (5<=zz), not a

Результатом логических операций является булева переменная, значение которой определяется по правилам согласно следующей таблицы истинности:

Значения булевых переменных		Результат логической операции			
a	b	a **and** b	a **or** b	**not** a	**not** b
true	true	true	true	false	false
true	false	false	true	false	true
false	true	false	true	true	false
false	false	false	false	true	true

Операторы условий if-then и **if-then-else.** Выполнение того или иного действия в программе может зависеть от какого-то условия или условий. Для этого используются операторы условий **if-then** и **if-then-else** со следующим форматом:

if условное выражение
then действие**;**

if условное выражение
then действие
else действие **;**

Действие операторов заканчивается точкой с запятой. Если при выполнении данного условия требуется выполнить несколько действий, то они начинаются со служебного слова **begin** и заканчиваются служебным словом **end**:

if условное выражение
then begin
действие 1, действие 2,...
end;

if условное выражение
then begin
действие 1, действие 2,...**end**
else begin
действие 1, действие 2,...**end;**

Пример. Требуется вычислить величину y, которая определяется следующими соотношениями:

$$y = \begin{cases} x^2, & x > 0; \\ \sin(x), & x \le 0. \end{cases}$$

if x>0
then y:=x*x
else y:=sin(x)**;**

Пример. При выполнении условия i≥2 вычислить значение y=0.5/z, значение x=tg(z+1), а затем перейти на оператор с меткой m1.

if i>=2
then begin
y:=0.5/z;
x:=tan(z+1.);
go to m1
end;

Операторы цикла for-to-do, **for-downto do**, **while-do.**

a) простые циклы **for-to-do** и **for-downto do** позволяют организовать повторяющиеся действия, количество которых определяется счетчиком цикла (целой переменной), начальное и конечное значение которого задается в операторе цикла. После выполнения одного цикла значение счетчика в цикле **for-to-do** увеличивается на единицу, а в цикле **for-downto do** уменьшается на единицу. Когда значение счетчика цикла выйдет за пределы конечного значения, цикл вычислений заканчивается. Если внутри цикла требуется выполнить несколько действий, то выделяется

блок таких действий с помощью служебных слов **begin** и **end**. Циклы **for-to-do** и **for-downto do** имеют следующий формат:

for имя счетчика**:=**начальное значение **to** конечное значение **do**

действие**;**

for имя счетчика**:=**начальное значение **downto** конечное значение **do**

begin

действие 1;

.....

действие n;

end;

Действие оператора заканчивается точкой с запятой.

Примеры:

for i:=-1 to 30 **do**

begin

y:=4+z;

x:=atan(y/z);

end;

for i:=400 **downto** k **do**

sqrt:=(4+x)/(a+exp(x))**;**

б) цикл **while-do** используется для организации цикла вычислений при выполнении какого-то условия (когда условное выражение принимает значение “истина” (**true**). Он имеет следующий формат:

while условное выражение **do**

действие**;**

while условное выражение **do**

begin

действие1;

.....

действие n;

```
end;
```

Действие заканчивается точкой с запятой.

Пример:

```
x:=1;
while x<100 do
begin
x:=x*5;
y=cos(x+3);
end;
```

ТЕМЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

В рамках изучения студентами дисциплины «Вычислительная математика» предусматривается выполнение самостоятельной работы по следующим темам:

- Математическое моделирование;
- Аппроксимация функций;
- Численное дифференцирование и интегрирование;
- Численные методы линейной алгебры;
- Решение нелинейных уравнений;
- Методы минимизации (оптимизации);
- Обыкновенные дифференциальные уравнения;
- Пакеты стандартных программ.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Амосов А.А. Вычислительные методы для инженеров. [Текст]: учебное пособие для втузов / А.А Амосов., Ю.А Дубинский., Н.В Копченова. - М.: Высшая школа, 2006. - 544 с.
2. Бахвалов Н.С. Численные методы [Текст]: учебное пособие /Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобольков. - М.: Лаборатория базовых знаний/БИНОМ, 2003.- 632 с.
3. Буторин, В.М. Численные методы [Текст]: ученое пособие: [для студ., обучающихся по направлению подготовки бакалавров 01.03.02] /В.М. Буторин, Т.В. Алябьева, А.А. Черепанов; Юго-Зап. гос. ун-т. - Курск: ЮЗГУ, 2015. - 167с.
4. Вержбицкий В.М. Численные методы математической физики [Электронный ресурс]: учебное пособие / В.М.Вержбицкий. - М: Директ.-Медиа, 2013. - 212 с. – Режим доступа: biblioclub.ru.
5. Волков Е.А. Численные методы [Текст]: учебное пособие / Е.А. Волков. - 4 –е изд., стер.– СПб.: Лань., 2007. - 256 с.
6. Демидович Б.П. Основы вычислительной математики [Текст]: учебное пособие для втузов /Б.П. Демидович, И.А Марон. - М.: Лань, 2006. - 672 с.
7. Самарский А.А. Введение в численные методов [Текст]: учебное пособие для студент. вузов /А.А. Самарский. - М.: Лань, 2005. - 288 с.
8. Турчак Л.И. Основы численных методов [Текст]: учебное пособие для вузов. /Л.И Турчак. - М.: Физматлит, 2005. - 304 с.