

```
>> fl=inline('0.02*x.*cos(0.01*x).^2')
```

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Курс лекций

- Уравнения и системы уравнений
- Задачи интерполяции и аппроксимации
- Фурье-анализ
- Численное дифференцирование и интегрирование
- Обыкновенные дифференциальные уравнения
- Дифференциальные уравнения в частных производных
- Интегральные уравнения

$$\begin{cases} y_0 = ax_0^2 + bx_0 + c \\ y_1 = ax_1^2 + bx_1 + c \\ y_2 = ax_2^2 + bx_2 + c \end{cases}$$

$$\left(\frac{1}{n} \sum x_i^4\right)a + \left(\frac{1}{n} \sum x_i^3\right)b + \left(\frac{1}{n} \sum x_i^2\right)c = \frac{1}{n} \sum y_i$$

```
function z=Simplex(A,b)  
'spline'
```

Сергей Поршнеv

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

Курс лекций

Допущено учебно-методическим объединением вузов по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений

Санкт-Петербург

«БХВ-Петербург»

2004

УДК 681.3.06+519.6
ББК 32.973Я73
П159

Поршнев С. В.

П159 Вычислительная математика. Курс лекций. — СПб.: БХВ-Петербург, 2004. — 320 с.: ил.

ISBN 5-94157-400-2

Книга представляет собой расширенный вариант лекций по курсу «Вычислительная математика», прочитанных автором в Нижнетагильском технологическом институте Уральского государственного технического университета для специальности «Информатика и вычислительная техника». Материал укладывается в перечень требований обязательного минимума содержания Государственного образовательного стандарта высшего профессионального образования по специальности 654600 «Информатика и вычислительная техника» дисциплины «Вычислительная математика». Основная особенность курса — прикладная направленность. Для каждого описанного в книге вычислительного метода приведены его программные реализации, а также соответствующие функции математического пакета MATLAB. Выбранный подход позволяет сформировать понимание математического содержания конкретного метода (границы его применимости, погрешности метода и т. д.) и умение использовать современные программные средства.

Для студентов и преподавателей профильных специальностей

УДК 681.3.06+519.6
ББК 32.973Я73

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Екатерина Капальгина</i>
Компьютерная верстка	<i>Екатерины Трубниковой</i>
Корректор	<i>Виктория Голуб</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 23.10.03.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 25,8.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-400-2

© Поршнев С. В., 2004

© Оформление, издательство "БХВ-Петербург", 2004

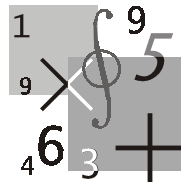
Содержание

Лекция 1. Теория погрешностей	1
1.1. Общие сведения об источниках погрешностей, их классификация	1
1.1.1. Виды погрешностей	1
1.2. Абсолютная и относительная погрешности. Формы записи данных	3
1.3. Вычислительная погрешность	5
1.4. Понятия о погрешности машинных вычислений	6
Лекция 2. Решение уравнений с одной переменной	11
2.1. Общие сведения и основные определения	11
2.2. Отделение корней	12
2.3. Метод половинного деления	12
2.4. Метод простой итерации	17
2.5. Преобразование уравнения к итерационному виду	20
Лекция 3. Методы решения систем линейных алгебраических уравнений	27
3.1. Общие сведения и основные определения	27
3.2. Метод Гаусса и его реализация в пакете MATLAB	28
3.3. Вычисление определителей	31
3.4. Решение систем линейных уравнений методом простой итерации	33
3.5. Метод Зейделя	39
3.6. Решение систем линейных уравнений средствами пакета MATLAB	42
Лекция 4. Методы решения систем нелинейных уравнений	45
4.1. Векторная запись нелинейных систем. Метод простых итераций	45
4.2. Метод Ньютона решения систем нелинейных уравнений	48
4.3. Решение нелинейных систем методами спуска	52
4.3.1. Метод Ньютона	72
4.4. Решение систем нелинейных уравнений средствами пакета MATLAB	80
Лекция 5. Интерполирование функций	85
5.1. Постановка задачи	85
5.2. Интерполяционный полином Лагранжа	89
5.3. Интерполяционный полином Ньютона для равноотстоящих узлов	91

5.3.1. Конечные разности	91
5.3.2. Первая интерполяционная формула Ньютона	92
5.3.3. Вторая интерполяционная формула Ньютона	95
5.4. Погрешность интерполяции	98
5.5. Сплайн-интерполяция	98
5.6. Решение задачи одномерной интерполяции средствами пакета MATLAB	104
Лекция 6. Численное дифференцирование и интегрирование	107
6.1. Численное дифференцирование функций, заданных аналитически	107
6.2. Особенности задачи численного дифференцирования функций, заданных таблицей	110
6.3. Интегрирование функций, заданных аналитически	111
6.4. Погрешность численного интегрирования	118
6.5. Вычисление интегралов методом Монте-Карло	121
Лекция 7. Методы обработки экспериментальных данных	123
7.1. Метод наименьших квадратов	123
7.2. Нахождение приближающей функции в виде линейной функции и квадратичного трехчлена	126
7.3. Аппроксимация функцией произвольного вида	134
Лекция 8. Преобразование Фурье	137
8.1. Разложение периодических функций в ряд Фурье	137
8.2. Эффект Гиббса	140
8.3. Спектральный анализ дискретных функций конечной длительности	145
8.4. Быстрое преобразование Фурье	146
Лекция 9. Численные методы решения обыкновенных дифференциальных уравнений	151
9.1. Общие сведения и определения	151
9.2. Метод Пикара	154
9.3. Метод Эйлера	156
9.4. Метод Рунге-Кутты	161
9.5. Средства пакета MATLAB для решения обыкновенных дифференциальных уравнений	167
Лекция 10. Численные методы решения дифференциальных уравнений в частных производных	171
10.1. Общие сведения и классификация уравнений в частных производных	171
10.2. Численные методы решения эллиптических уравнений	174
10.3. Явные разностные схемы для уравнений параболического и эллиптического типов	182
10.4. Неявная разностная схема для уравнения параболического типа	187
10.5. Решение уравнений с частными производными методом Монте-Карло	192
Лекция 11. Численные методы решения интегральных уравнений	199
11.1. Общие сведения об интегральных уравнениях	199
11.2. Квадратурный метод решения интегральных уравнений Фредгольма	205
11.3. Квадратурный метод решения интегральных уравнений Вольтерра	210

Приложение 1. Основные приемы работы с пакетом matlab	221
П1.1. Работа в командном окне.....	221
П1.1.1. Вход в систему MATLAB.....	222
П1.1.2. Интерактивный доступ к справочной информации и документации	222
П1.1.2.1. Команда <i>help</i>	222
П1.1.2.2. Команда <i>lookfor</i>	223
П1.1.2.3. Меню <i>Help</i>	224
П1.1.3. Редактирование и повторный вызов командной строки	225
П1.1.4. Формат вывода	225
П1.1.5. Копия протокола текущей сессии	226
П1.2. Создание матриц	226
П1.2.1. Явное задание матриц	227
П1.2.2. Подматрицы и использование двоеточия.....	228
П1.2.3. Функции построения матриц.....	229
П1.3. Операции, выражения и переменные	230
П1.3.1. Правила записи операторов.....	230
П1.3.2. Матричные операции	231
П1.3.3. Операции с массивами	232
П1.3.4. Сохранение данных из рабочей области	232
П1.4. Операторы <i>for</i> , <i>while</i> , <i>if</i> , <i>case</i> и операторы отношения.....	232
П1.4.1. Цикл <i>for</i>	232
П1.4.2. Цикл <i>while</i>	233
П1.4.3. Условный оператор <i>if</i>	234
П1.4.4. Оператор переключения <i>case</i>	234
П1.4.5. Условия (операторы отношения)	235
П1.5. Функции MATLAB	235
П1.5.1. Скалярные функции	235
П1.5.2. Векторные функции	236
П1.5.3. Матричные функции	236
П1.6. М-файлы.....	237
П1.6.1. Файлы-программы.....	237
П1.6.2. Файлы-функции.....	238
П1.6.3. Текстовые строки, сообщения об ошибках	241
П1.6.4. Работа с m-файлами	241
П1.6.5. Список путей доступа	242
П1.6.5.1. Работа со списком путей доступа	242
П1.6.5.2. Текущий каталог.....	243
П1.6.5.3. Средство просмотра и редактирования путей доступа Path Browser.....	243
П1.6.5.4. Использование редактора/отладчика	244
П1.6.6. Отладка m-файлов	246
Приложение 2. Точные методы решения дифференциальных уравнений в частных производных	249
П2.1. Метод разделения переменных	249
П2.2. Метод интегральных преобразований	253
П2.2.1. Общие сведения об интегральных преобразованиях	253
П2.2.2. Решение краевой задачи УЧП с использованием синус-преобразования	255

П2.2.2.3. Решение краевой задачи уравнения в частных производных с использованием преобразования Фурье.....	259
П2.2.2.4. Решение краевой задачи УЧП методом преобразования Лапласа	261
П2.3. Метод преобразования зависимых переменных.....	263
П2.4. Метод преобразования координат	264
П2.5. Метод разложения по собственным функциям	267
П2.6. Метод функций Грина	271
П2.7. Метод интегральных уравнений	277
Приложение 3. Приближенные методы решения дифференциальных уравнений в частных производных	283
ПЗ.1. Вариационный метод	283
ПЗ.2. Методы теории возмущений	286
ПЗ.2.1. Применение метода последовательных приближений к решению обыкновенных нелинейных дифференциальных уравнений	286
ПЗ.2.1. Обычная формула теории возмущений для решения нелинейных дифференциальных уравнений	288
Приложение 4. Метод обратной задачи рассеивания	297
Литература	303



Лекция 1

Теория погрешностей

Цель лекции: изложить общие сведения об источниках и видах погрешностей, правилах вычисления абсолютной и относительной погрешностей, формах записи чисел, познакомить со способами хранения действительных чисел в памяти компьютера и ввести понятие о конечной точности машинной арифметики.

1.1. Общие сведения об источниках погрешностей, их классификация

Источниками возникновения погрешности численного решения задачи являются:

- неточность математического описания, в частности, неточность задания начальных данных;
- неточность численного метода решения задачи, возникающая, например, когда решение математической задачи требует неограниченного или неприемлемо большого числа арифметических операций, что приводит к необходимости ограничения их числа, т. е. использования приближенного решения;
- конечная точность машинной арифметики.

1.1.1. Виды погрешностей

Неустраняемая погрешность состоит из двух частей:

- погрешности, обусловленной неточностью задания числовых данных, входящих в математическое описание задачи;
- погрешности, являющейся следствием несоответствия математического описания задачи реальной действительности (погрешность математической модели).

Для вычислителя погрешность задачи следует считать неустранимой, хотя постановщик задачи иногда может ее изменить.

Результирующая погрешность определяется как сумма величин всех перечисленных выше погрешностей.

Погрешность метода связана со способом решения поставленной математической задачи. Она появляется в результате замены исходной математической модели другой и/или конечной последовательностью других более простых (например, линейных) моделей. При создании численных методов закладывается возможность отслеживания таких погрешностей и доведения их до сколь угодно малого уровня. Отсюда естественно отношение к погрешности метода как устранимой (или условной).

Вычислительная погрешность (погрешность округлений) обусловлена необходимостью выполнять арифметические операции над числами, усеченными до количества разрядов, зависящего от применяемой вычислительной техники.

Рассмотрим простой пример, иллюстрирующий описанные ранее виды погрешностей, на основе задачи описания движения маятника (рис. 1.1), в которой требуется предсказать угол отклонения маятника от вертикали Θ , начинающего движение в момент времени $t = t_0$.

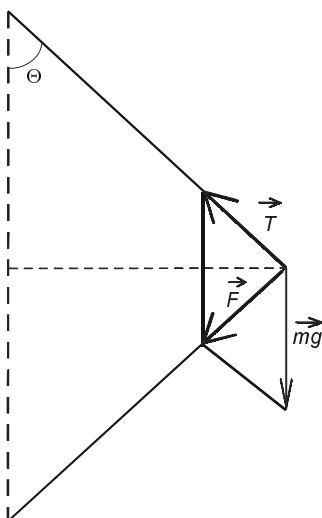


Рис. 1.1. Модель математического маятника

Движение маятника может быть описано дифференциальным уравнением второго порядка:

$$l \frac{d^2\theta}{dt^2} + g \sin \theta + \mu \frac{d\theta}{dt} = 0, \quad (1.1)$$

где l — длина маятника, g — ускорение свободного падения, μ — коэффициент трения.

Причины возникновения погрешностей в данной задаче могут быть различными.

- Реальная сила трения зависит от скорости движения маятника по нелинейному закону.
- Значения величин l , g , μ , t_0 , $y \pm e_y$, $\theta'(t_0)$ известны с некоторыми погрешностями.
- Для решения уравнения (1.1), не имеющего аналитического решения, приходится использовать численный метод, вследствие чего возникает погрешность метода.
- Вычислительная погрешность, возникающая вследствие конечной точности представления чисел в компьютере.

1.2. Абсолютная и относительная погрешности. Формы записи данных

Определение 1.1. Если a — точное значение некоторой величины и a^* — известное приближение к нему, то абсолютной погрешностью приближенного значения a^* называют некоторую величину $\Delta(a^*)$, про которую известно, что

$$|a^* - a| \leq \Delta(a^*). \quad (1.2)$$

Определение 1.2. Относительной погрешностью приближенного значения называют некоторую величину $\delta(a^*)$, про которую известно, что

$$\left| \frac{a^* - a}{a^*} \right| \leq \delta(a^*). \quad (1.3)$$

Относительную погрешность часто выражают в процентах.

Определение 1.3. Значащими цифрами числа называют все цифры в его записи, начиная с первой ненулевой слева.

Пример 1.1.

$$a^* = 0.03045 \quad a^* = 0.03045000^1$$

¹ При записи чисел в качестве знака, разделяющего целую и дробную часть числа, мы используем, как это принято в программировании, точку.

Примечание

Здесь цифры, записанные курсивом, значащие.

Определение 1.4. Значащую цифру называют верной, если модуль погрешности числа не превосходит единицы разряда, соответствующего этой цифре.

Пример 1.2.

$$\begin{aligned} a^* &= 0.0\mathbf{3045} & \Delta(a^*) &= 0.000003 \\ a^* &= 0.0\mathbf{30450000} & \Delta(a^*) &= 0.00000007 \end{aligned}$$

Примечание

Здесь цифры, записанные курсивом, верные.

Определение 1.5. Число записано со всеми верными цифрами, если в его записи представлены только верные значащие цифры.

Иногда употребляется термин *число верных цифр после запятой*: подсчитывается число верных цифр после запятой от первой цифры до последней верной цифры.

Довольно часто информация о некоторой величине задается пределами измерений

$$a_1 \leq a \leq a_2 .$$

Принято записывать эти пределы с одинаковым числом знаков после запятой, так как обычно достаточно грубого представления о погрешности. В записи чисел a_1 , a_2 обычно берут столько значащих цифр, сколько нужно для того, чтобы разность $a_2 - a_1$ содержала одну, две значащие цифры.

Информацию о том, что a^* является приближенным значением числа a с абсолютной погрешностью $\Delta(a^*)$, принято также записывать в виде:

$$a = a^* \pm \Delta(a^*) . \quad (1.4)$$

Числа a^* , $\Delta(a^*)$ принято записывать с одинаковым числом знаков после запятой.

Пример 1.3.

$$\left\{ \begin{array}{l} a = 1.123 \pm 0.004 \\ a = 1.123 \pm 4 \cdot 10^{-3} \\ 1.123 - 0.004 \leq a \leq 1.123 + 0.004 \end{array} \right.$$

Информацию о том, что a^* является приближенным значением числа a с относительной погрешностью $\delta(a^*)$, записывают в виде:

$$a = a^* (1 \pm \delta(a^*)).$$

Пример 1.4.

$$1.123 \cdot (1 - 0.003) \leq a \leq 1.123 \cdot (1 + 0.003)$$

Примечание

Данная запись числа эквивалентна записи чисел из примера 1.3.

1.3. Вычислительная погрешность

Далее для краткости будем обозначать абсолютную погрешность числа x как e_x , относительную погрешность — δ_x .

□ Погрешность суммирования чисел $x \pm e_x$, $y \pm e_y$:

- абсолютная погрешность

$$z = (x \pm e_x) + (y \pm e_y) = (x + y) \pm (e_x + e_y);$$

- относительная погрешность

$$\delta_z = \frac{e_x + e_y}{|x + y|} = \frac{e_x}{|x + y|} \frac{|x|}{|x|} + \frac{e_y}{|x + y|} \frac{|y|}{|y|} = \frac{|x|}{|x + y|} \delta_x + \frac{|y|}{|x + y|} \delta_y.$$

□ Погрешность вычитания чисел $x \pm e_x$, $y \pm e_y$:

- абсолютная погрешность

$$z = (x \pm e_x) - (y \pm e_y) = (x - y) \pm (e_x + e_y);$$

- относительная погрешность

$$\delta_z = \frac{e_x + e_y}{|x - y|} = \frac{e_x}{|x - y|} \frac{|x|}{|x|} + \frac{e_y}{|x - y|} \frac{|y|}{|y|} = \frac{|x|}{|x - y|} \delta_x + \frac{|y|}{|x - y|} \delta_y.$$

□ Погрешность умножения чисел $x \pm e_x$, $y \pm e_y$:

- абсолютная погрешность

$$z = (x \pm e_x) \cdot (y \pm e_y) = x \cdot y \pm y \cdot e_x \pm x \cdot e_y + e_x \cdot e_y \approx x \cdot y \pm y \cdot e_x \pm x \cdot e_y;$$

- относительная погрешность

$$\delta_z = \frac{|y| \cdot e_x + |x| \cdot e_y}{|x \cdot y|} = \frac{e_x}{|x|} + \frac{e_y}{|y|} = \delta_x + \delta_y.$$

□ Погрешность деления чисел $x \pm e_x$, $y \pm e_y$:

- абсолютная погрешность

$$z = \frac{x \pm e_x}{y \pm e_y} = \frac{(x \pm e_x) \cdot (y \pm e_y)}{(y \pm e_y) \cdot (y \pm e_y)} \approx \frac{x}{y} \pm \frac{y \cdot e_x + x \cdot e_y}{y^2};$$

- относительная погрешность

$$\delta_z = \frac{|y| \cdot e_x + |x| \cdot e_y}{\left| \frac{x}{y} \right|} = \frac{|y| \cdot e_x + |x| \cdot e_y}{y^2 \cdot \left| \frac{x}{y} \right|} = \frac{e_x}{|x|} + \frac{e_y}{|y|} = \delta_x + \delta_y.$$

□ Погрешность функции, зависящей от одной переменной:

- абсолютная погрешность

$$f(x \pm e_x) \approx f(x) \pm f'(x) \cdot e_x,$$

$$\Delta f = f(x \pm e_x) - f(x) = |f'(x)| e_x;$$

- относительная погрешность

$$\left| \frac{\Delta f}{f} \right| = \left| \frac{f'(x)}{f(x)} \right| e_x.$$

Аналогично получают формулы для оценки абсолютной и относительной погрешностей для функций, зависящих от n переменных.

1.4. Понятия о погрешности машинных вычислений

Для представления чисел в памяти компьютера применяют два способа.

□ С фиксированной запятой².

□ С плавающей запятой.

Пусть в основу запоминающего устройства положены однотипные физические устройства, имеющие r устойчивых состояний, называемых регистрами. Причем каждому устройству ставится в соответствие одинаковое количество k регистров и, кроме того, с помощью регистров может фиксироваться знак. Упорядоченные элементы образуют разрядную сетку машинного слова: в каждом разряде может быть записано одно из базисных чисел $0, 1, \dots, r - 1$ (одна из r "цифр" r -ой системы исчисления) и в специаль-

² Здесь в обозначении устройств хранения информации мы используем устоявшуюся в отечественной литературе терминологию, в которой в качестве разделительного знака между целой и дробными частями числа используется запятая.

ном разряде отображен знак "+" или "-". При записи чисел с фиксированной запятой кроме упомянутых r параметров (основания системы счисления) и k (количество разрядов, отводимых под запись числа) указывается еще общее количество l разрядов, выделяемых под дробную часть числа. Таким образом, положительное вещественное число a , представляющее собой в r -ой системе бесконечную, непериодическую дробь, отображается конечной последовательностью

$$\alpha_1 \alpha_2 \dots \alpha_{k-l} \alpha_{k-l+1} \dots \alpha_{k-1} \alpha_k,$$

где $\alpha_i \in \{0; 1; \dots; r-1\}$, т. е.

$$a^* \approx \alpha_1 r^{k-l-1} + \alpha_2 r^{k-l-2} + \dots + \alpha_{k-l} r^0 + \alpha_{k-l+1} r^{-1} + \dots + \alpha_{k-1} r^{-(l-1)} + \alpha_k r^{-l}$$

Диапазон представляемых таким способом чисел определяется числами с наибольшими цифрами во всех разрядах, т. е. наименьшим числом $-(r-1)(r-1)\dots(r-1)$ и наибольшим $(r-1)(r-1)\dots(r-1)$, а абсолютная точность представления — есть оценка величины $|a - a^*|$, зависящая от способа округления. Абсолютная точность представления вещественных чисел с фиксированной запятой одинакова в любой части диапазона. В то же время относительная точность представления числа может значительно различаться в зависимости от того, берется ли a близким к нулю или к границе диапазона.

Пример 1.5. Рассмотрим запоминающее устройство с фиксированной запятой, состоящее из $k = 7$ элементов и имеющее $r = 10$ ($r = 0, 1, \dots, 9$). Будем считать, что общее количество разрядов, выделяемых под дробную часть, $l = 3$, под знак — один разряд, под целую часть — три разряда (рис. 1.2).

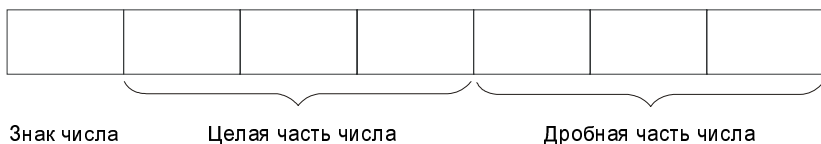


Рис. 1.2. Схема машинного слова запоминающего устройства с фиксированной запятой

Тогда наибольшее число, которое можно сохранить в данном запоминающем устройстве, равно 999.999, а наименьшее равно -999.999. У любого числа из указанного диапазона, являющегося бесконечной периодической дробью, вне зависимости от его величины после запятой сохраняется только 3 цифры. Поэтому абсолютная точность представления чисел $a_1 = 1.123456$ и $a_2 = 999.123456$ оказывается одинаковой.

$$\Delta_1 = 1.123456 - 1.123 = 0.000456,$$

$$\Delta_2 = 999.123456 - 999.123 = 0.000456.$$

Относительные погрешности представления этих чисел в запоминающем устройстве будут различны:

$$\delta_1 = \frac{\Delta_1}{a_1} \approx 0.04 \%,$$

$$\delta_2 = \frac{\Delta_2}{a_2} = 0.5 \cdot 10^{-4} \%.$$

В основе часто употребляемого представления **с плавающей запятой** лежит экспоненциальная форма записи числа:

$$a = M \cdot r^p,$$

где r — основание, p — порядок, M — мантисса ($r^{-1} \leq |M| \leq 1$). Если под мантиссу выделяется l r -ичных элементов, а под порядок — m , то в системе записи с плавающей запятой вещественное число a представляется конечным числом

$$a^* \approx \pm (\beta_1 r^{-1} + \beta_2 r^{-2} + \dots + \beta_l r^{-l}) \cdot r^\gamma,$$

где γ — целое число из промежутка $[-r^m, r^m - 1]$; $\beta_i \in \{1; \dots; r - 1\}$, $i = 2, \dots, l$.

Структура машинного слова запоминающего устройства с плавающей запятой представлена на рис. 1.3.

Знак порядка	Порядок числа (m разрядов)	Знак мантиссы	Мантисса числа (l разрядов)
--------------	-------------------------------	---------------	--------------------------------

Рис. 1.3. Схема машинного слова запоминающего устройства с плавающей запятой

Числа $\pm r^{\pm m}$ определяют границы допустимого числового диапазона.

Относительная точность представления вещественных чисел равна

$$\left| \frac{a - a^*}{a} \right| = \frac{\beta_{l+1} r^{-(l+1)} + \beta_{l+2} r^{-(l+2)} + \dots}{\beta_1 r^{-1} + \beta_2 r^{-2} + \dots} \leq \frac{r^{-l}}{\beta_1 \cdot r^{-1}} \leq r^{1-l},$$

т. е. относительная точность одинакова в любой части числового диапазона и зависит лишь от числа разрядов, отводимых под мантиссу числа.

Например, для записи числа в 48-разрядном машинном слове БЭСМ-6 40 двоичных разрядов выделялись под мантиссу, 6 — под порядок числа и 2 — под знаки мантиссы, т. е. $r = 2$, $l = 40$, $m = 6$. Следовательно, точность представления чисел с плавающей запятой не хуже 2^{-39} ($\approx 10^{-12}$), граница машинного нуля 2^{-64} ($\approx 10^{-19}$), машинной бесконечности 2^{63} .

Пример 1.6. Рассмотрим запоминающее устройство, состоящее из $k = 8$ элементов и имеющее $r = 10$ ($r = 0, 1, \dots, 9$). Будем считать, что общее количество разрядов, выделяемых под дробную часть $l = 5$, под знак порядка числа — 1 ячейка, под знак мантиссы числа — 1 ячейка, под порядок — 2 ячейки.

Оценим точность представления чисел $a_1 = 1.123123$ и $a_2 = 999.123123$.

Запишем числа в форме представления с плавающей запятой:

$$a_1 = 0.1123123 \cdot 10^1,$$

$$a_2 = 0.999123123 \cdot 10^3.$$

В запоминающем устройстве числа a_1 , a_2 будут записаны в виде, показанном на рис. 1.4.

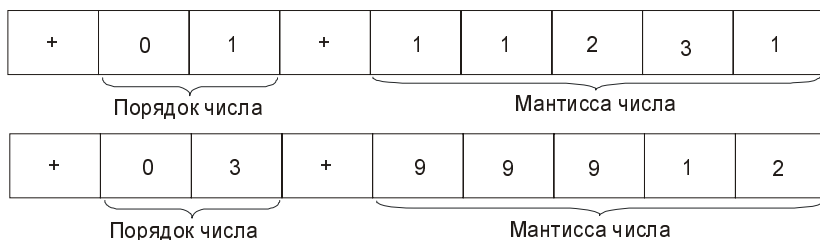


Рис. 1.4. Представление чисел a_1 , a_2 с плавающей запятой в запоминающем устройстве

Абсолютные погрешности чисел равны:

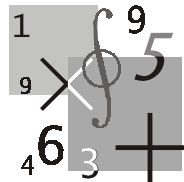
$$\Delta_1 = (0.1123123 - 0.11231) \cdot 10^1 \approx (0.23 \cdot 10^{-5}) \cdot 10^1$$

$$\Delta_2 = (0.999123123 - 0.99912) \cdot 10^3 \approx (0.31 \cdot 10^{-5}) \cdot 10^3.$$

Относительные погрешности чисел равны:

$$\delta_1 = \frac{\Delta_1}{a_1} = \frac{(0.23 \cdot 10^{-5}) \cdot 10^1}{0.1123 \cdot 10^1} \approx 2 \cdot 10^{-3} \%,$$

$$\delta_2 = \frac{\Delta_2}{a_2} = \frac{(0.31 \cdot 10^{-5}) \cdot 10^3}{0.99912 \cdot 10^3} \approx 3 \cdot 10^{-3} \%.$$



Лекция 2

Решение уравнений с одной переменной

Цель лекции: на примере методов половинного деления и простой итерации продемонстрировать основные подходы к решению задачи нахождения корней нелинейных уравнений, описать соответствующие алгоритмы и их программные реализации, обсудить использование соответствующих функций пакета MATLAB.

2.1. Общие сведения и основные определения

Наиболее общий вид нелинейного уравнения:

$$F(x) = 0, \quad (2.1)$$

где функция $F(x)$ определена и непрерывна на конечном или бесконечном интервале $[a, b]$.

Определение 2.1. Всякое число $\xi \in [a, b]$, обращающее функцию $F(x)$ в нуль, называется корнем уравнения (2.1).

Определение 2.2. Число ξ называется корнем k -ой кратности, если при $x = \xi$ вместе с функцией $F(x)$ равны нулю ее производные до $(k-1)$ -го порядка включительно:

$$F(\xi) = F'(\xi) = \dots = F^{(k-1)}(\xi) = 0. \quad (2.2)$$

Определение 2.3. Однократный корень называется простым.

Определение 2.4. Уравнения $F(x)=0$ и $G(x)=0$ называются равносильными (эквивалентными), если множества решений данных уравнений совпадают.

Нелинейные уравнения с одной переменной подразделяются на *алгебраические* и *трансцендентные*.

Определение 2.5. Уравнение (2.1) называется алгебраическим, если функция $F(x)$ является алгебраической.

Путем алгебраических преобразований из всякого алгебраического уравнения можно получить уравнение в канонической форме:

$$P_n(x) = a_0x^n + a_1x^{n-1} + \dots + a_n, \quad (2.3)$$

где a_0, a_1, \dots, a_n — действительные коэффициенты уравнения, x — неизвестное.

Из алгебры известно, что всякое алгебраическое уравнение имеет, по крайней мере, один вещественный или два комплексно сопряженных корня.

Определение 2.6. Уравнение (2.1) называется трансцендентным, если функция $F(x)$ не является алгебраической.

Определение 2.7. Решить уравнение (2.1) означает:

1. Установить, имеет ли уравнение корни.
2. Определить число корней уравнения.
3. Найти значения корней уравнения с заданной точностью.

2.2. Отделение корней

Определение 2.8. Отделение корней — процедура нахождения отрезков, на которых уравнение (2.1) имеет только одно решение.

В большинстве случаев отделение корней можно провести графически. Для этого достаточно построить график функции $F(x)$ и определить отрезки, на которых эта функция имеет только одну точку пересечения с осью абсцисс.

В сомнительных случаях графическое отделение корней необходимо подкреплять вычислениями. При этом можно использовать следующие очевидные положения:

- если непрерывная функция принимает на концах отрезка $[a, b]$ значения разных знаков (т. е. $F(a) \cdot F(b) < 0$), то уравнение (2.1) имеет на этом отрезке по меньшей мере один корень;
- если функция $F(x)$ к тому же и строго монотонна, то корень на отрезке единственный.

2.3. Метод половинного деления

Пусть уравнение (2.1) имеет на отрезке $[a, b]$ единственный корень, причем функция $F(x)$ на данном отрезке непрерывна (рис. 2.1).

Разделим отрезок $[a, b]$ пополам точкой $c = (a+b)/2$. Если $F(c) \neq 0$, то возможны два случая:

1. Функция $F(x)$ меняет знак на отрезке $[a, c]$.
2. Функция $F(x)$ меняет знак на отрезке $[c, b]$.

Выбирая в каждом случае тот отрезок, на котором функция меняет знак, и продолжая процесс половинного деления дальше, можно прийти к сколь угодно малому отрезку, содержащему корень уравнения.

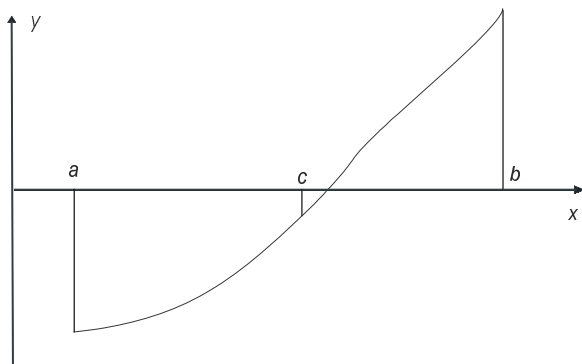


Рис. 2.1. К объяснению метода половинного деления

Пример 2.1. Найти, используя пакет MATLAB, методом половинного деления корень уравнения $x^4 - 11x^3 + x^2 + x + 0.1 = 0$.

1. Создайте файл `Func.m` (листинг 2.1), содержащий описание функции $f(x) = x^4 - 11x^3 + x^2 + x + 0.1$.

Листинг 2.1. Файл `Func.m`

```
function z=Func(x)
z=x.^4-11*x.^3+x.^2+x+0.1;
```

2. Создайте файл `Div2.m` (листинг 2.2), содержащий описание функции, возвращающей значение корня уравнения $f(x) = 0$ методом половинного деления.

Листинг 2.2. Файл `Div2.m`

```
function z=Div2(f,x1,x2,eps);
% f - имя m-файла, содержащего описание функции f(x)
% x1 - левая граница отрезка, на котором производится поиск решения
% уравнения
% x2 - правая граница отрезка, на котором производится поиск решения
```

```

% уравнения
% eps – точность решения
L=x2-x1;
while L>eps
    c=(x2+x1)/2;
    if feval(f,c)*feval(f,x1)<0
        % feval(f,c) – оператор вычисления в точке x=c значения функции,
        % описание которой находится в соответствующем файле.
        % Имя файла хранится в строковой переменной f
        x2=c;
    else
        x1=c;
    end;
    L=x2-x1;
end;
z=c;

```

Примечание

Обращаем внимание читателя, что в пакете MATLAB существует "проблема буквы "я", суть которой состоит в том, что код строчной буквы "я" русского алфавита рассматривается пакетом MATLAB, как код команды. Поэтому обнаружение прописной буквы "я" в комментарии приводит к останову выполняемого файла и появлению сообщения об ошибке этапа исполнения. Для предотвращения описанной ситуации следует использовать в комментариях только строчную букву "я".

3. Постройте график функции на интервале $[-1, 1]$ (рис. 2.2), выполнив в командном окне пакета MATLAB следующую последовательность операторов:

```

>> x1=-1;
>> x2=1;
>> dx=10^-3;
>> x=x1:dx:x2;
>> plot(x,Func(x)); grid on

```

4. Вычислите значения корня уравнения:

```

>> Div2('Func',x1,x2,10^-5)
ans =
    0.3942

```

5. Проверьте полученное значение корня:

```

>> Func(ans)
ans =
    7.4926e-006

```

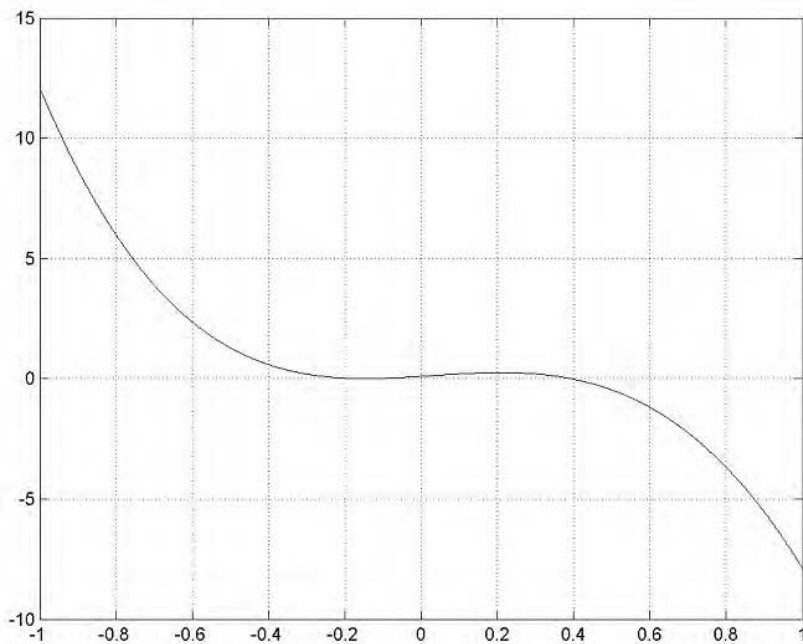


Рис. 2.2. График функции $f(x) = x^4 - 11x^3 + x^2 + x + 0.1$

Для рассмотрения процесса нахождения корня уравнения в динамике, необходимо сохранить значение корня на каждом шаге вычислительной процедуры и построить зависимость значения корня от номера шага. Далее приведен листинг файла Div2l.m, содержащего описание функции, возвращающей значение корня и длины отрезка (на котором данный корень находится) на каждом шаге метода половинного деления.

Листинг 2.3. Файл Div2l.m

```
function [z1,z2]=Div2(f,x1,x2,eps);
k=1;
L(1)=x2-x1; % начальная длина отрезка
c(1)=(x2+x1)/2; % начальное значение корня
while L(k)>eps
    if feval(f,c(k))*feval(f,x1)<0
        x2=c(k);
    else
        x1=c(k);
    end;
    k=k+1;
    c(k)=(x2+x1)/2;
```

```
L(k)=x2-x1;
```

```
end;
```

```
z1=c;
```

```
z2=L;
```

6. На каждом шаге итерационного процесса вычислите значения корня и длины отрезка, на котором производится поиск решения.

```
>> [c L]=Div2I('Func', x1, x2, 10^-5);
```

7. Визуализируйте зависимость значения корня от номера итерационного процесса (рис. 2.3).

```
>> plot(c, '-o')
```

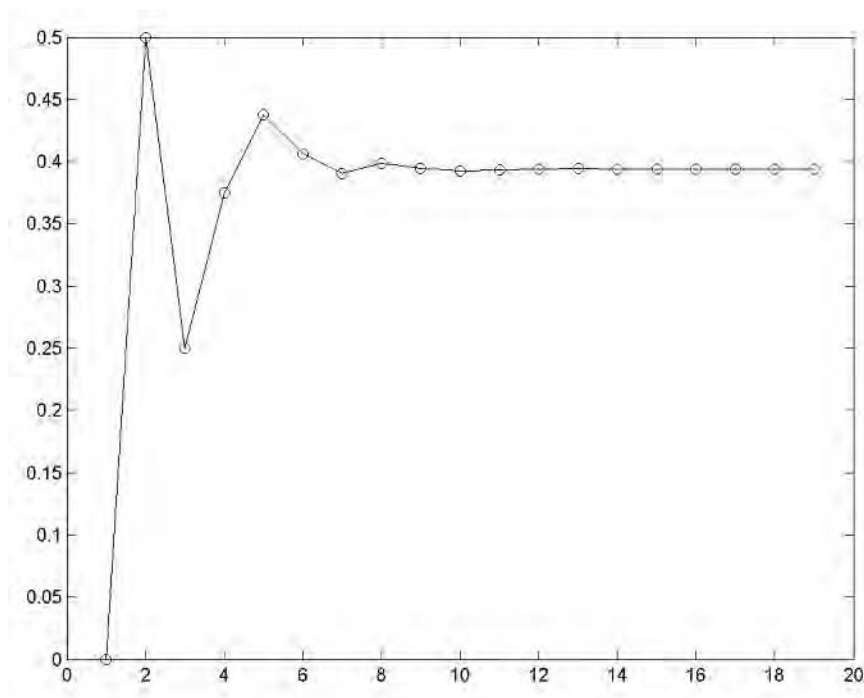


Рис. 2.3. Зависимость значения корня от номера шага вычислительной процедуры

8. Визуализируйте зависимость длины отрезка, на котором ищется значение корня, от номера итерации (рис. 2.4).

```
>> plot(L, '-o')
```

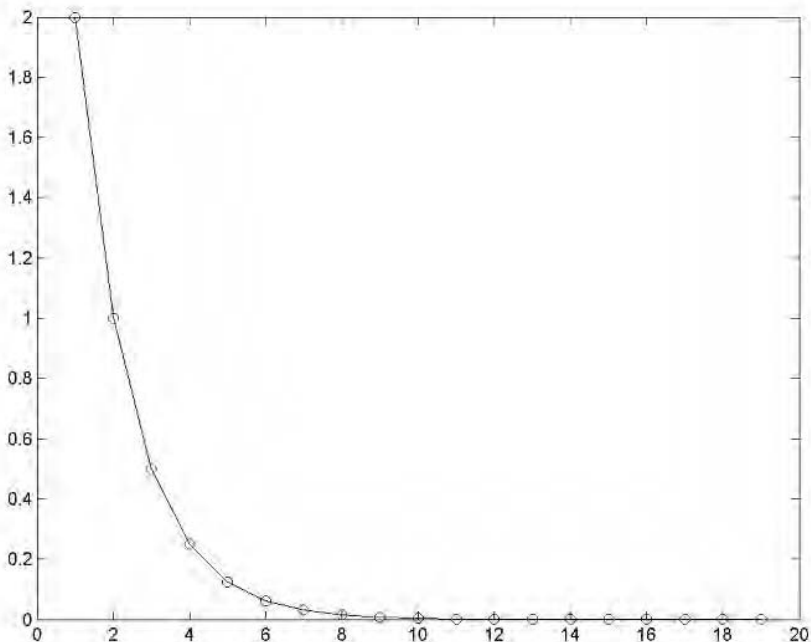


Рис. 2.4. Зависимость длины отрезка, на котором ищется значение корня, от номера шага вычислительной процедуры

2.4. Метод простой итерации

Заменяем уравнение (2.1) равносильным уравнением:

$$x = f(x) \quad (2.4)$$

Пусть ξ — корень уравнения (2.4), а x_0 — полученное каким-либо способом нулевое приближение к корню ξ . Подставляя x_0 в правую часть уравнения (2.4), получим некоторое число $x_1 = f(x_0)$. Повторим данную процедуру с x_1 и получим $x_2 = f(x_1)$. Повторяя описанную процедуру, получим последовательность

$$x_0, x_1, \dots, x_n, \dots, \quad (2.5)$$

называемую итерационной последовательностью.

Геометрическая интерпретация данного алгоритма представлена на рис. 2.5.

Итерационная последовательность, вообще говоря, может быть как сходящейся, так и расходящейся, что определяется видом функции $f(x)$.

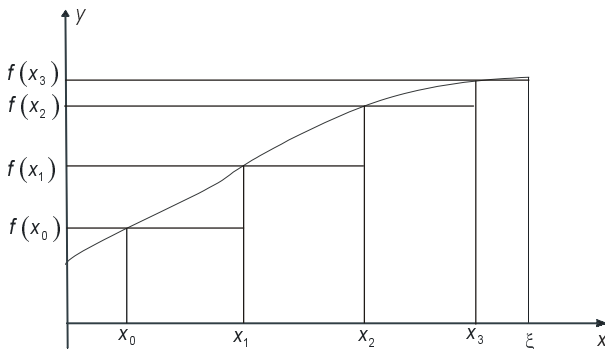


Рис. 2.5. К объяснению метода простой итерации

Теорема 2.1. Если функция $f(x)$ непрерывна, а последовательность (2.5) сходится, то предел последовательности (2.5) является корнем уравнения (2.4).

Действительно, пусть $\xi = \lim_{n \rightarrow \infty} x_n$. Перейдем к пределу в равенстве $x_n = f(x_{n-1})$:

$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} f(x_{n-1}) = f\left(\lim_{n \rightarrow \infty} x_{n-1}\right) = f(\xi). \quad (2.6)$$

Условие сходимости итерационного процесса определяется следующей теоремой.

Теорема 2.2. Достаточное условие сходимости итерационного процесса.

Пусть уравнение $x = f(x)$ имеет единственный корень на отрезке $[a, b]$ и выполнены условия:

1. $f(x)$ определена и дифференцируема на $[a, b]$.
2. $f(x) \in [a, b]$ для всех $x \in [a, b]$.
3. Существует такое вещественное q , что $|f'(x)| \leq q < 1$ для всех $x \in [a, b]$.

Тогда итерационная последовательность $x_n = f(x_{n-1})$ ($n = 1, 2, \dots$) сходится при любом начальном приближении $x_0 \in [a, b]$.

Доказательство. Построим итерационную последовательность вида (2.5) с любым начальным значением $x_0 \in [a, b]$. В силу условия 2 теоремы 2.2 все члены последовательности находятся в отрезке $[a, b]$.

Рассмотрим два последовательных приближения $x_n = f(x_{n-1})$ и $x_{n+1} = f(x_n)$. По теореме Лагранжа о конечных приращениях имеем:

$$x_{n+1} - x_n = f(x_n) - f(x_{n-1}) = f'(c)(x_n - x_{n-1}), c \in [x_{n-1}, x_n].$$

Переходя к модулям и принимая во внимание условие 3 теоремы 2.2, получим:

$$\begin{aligned} |x_{n+1} - x_n| &= |f'(c)| \cdot |x_n - x_{n-1}| \leq q|x_n - x_{n-1}|, \\ |x_{n+1} - x_n| &\leq q|x_n - x_{n-1}|. \end{aligned}$$

При $n = 1, 2$, имеем:

$$\begin{aligned} |x_2 - x_1| &\leq q|x_1 - x_0|, \\ |x_3 - x_2| &\leq q \cdot |x_2 - x_1| \leq q^2|x_1 - x_0|, \end{aligned} \quad (2.7)$$

$$|x_{n+1} - x_n| \leq q^n|x_1 - x_0|.$$

Рассмотрим ряд

$$x_0 + (x_1 - x_0) + (x_2 - x_1) + \dots + (x_n - x_{n-1}) + \dots \quad (2.8)$$

Составим частичные суммы этого ряда

$$S_1 = x_0, S_2 = x_1, \dots, S_{n+1} = x_n.$$

Заметим, что $(n+1)$ -я частичная сумма ряда (2.8) совпадает с n -ым членом итерационной последовательности (2.5), т. е.

$$S_{n+1} = x_n. \quad (2.9)$$

Сравним ряд (2.8) с рядом

$$|x_1 - x_0| + q|x_1 - x_0| + q^2|x_1 - x_0| + \dots \quad (2.10)$$

Заметим, что в силу соотношения (2.7) абсолютные величины членов ряда (2.8) не превосходят соответствующих членов ряда (2.10). Но ряд (2.10) сходится как бесконечно убывающая геометрическая прогрессия ($q < 1$, по условию). Следовательно, и ряд (2.8) сходится, т. е. его частичная сумма (2.9) имеет предел. Пусть $\xi = \lim_{n \rightarrow \infty} x_n$. В силу непрерывности функции f получаем (см. (2.6)):

$$\xi = f(\xi)$$

т.е. ξ — корень уравнения $x = f(x)$.

Отметим, что условия теоремы не являются необходимыми. Это означает, что итерационная последовательность может оказаться сходящейся и при невыполнении этих условий.

Отыщем погрешность корня уравнения, найденного методом простой итерации. Пусть x_n — приближение к истинному значению корня уравнения $x = f(x)$. Абсолютная ошибка приближения x_n оценивается модулем

$$\Delta x_n = |\xi - x_n|.$$

Принимая во внимание (2.8) и (2.9), имеем

$$\xi - x_n = \xi - S_{n+1} = (x_{n+1} - x_n) + (x_{n+2} - x_{n+1}) + \dots \quad (2.11)$$

Сравним (2.11) с остатком ряда (2.9):

$$q^n |x_1 - x_0| + q^{n+1} |x_1 - x_0| + \dots \quad (2.12)$$

Учитывая оценку (2.7), получаем

$$|\xi - x_n| \leq q^n |x_1 - x_0| + q^{n+1} |x_1 - x_0| + \dots = \frac{q^n}{1-q} |x_1 - x_0|.$$

Таким образом, для оценки погрешности n -го приближения получается формула

$$\Delta x_n \leq \frac{q^n}{1-q} |x_1 - x_0|. \quad (2.13)$$

На практике удобнее использовать модификацию формулы (2.13).

Примем за нулевое приближение x_{n-1} (вместо x_0). Следующим приближением будет x_n (вместо x_1). Так как $|x_n - x_{n-1}| \leq q^{n-1} |x_1 - x_0|$, то

$$\Delta x_n \leq \frac{q}{1-q} |x_n - x_{n-1}|. \quad (2.14)$$

При заданной точности ответа ε итерационный процесс прекращается, если $\Delta x_n \leq \varepsilon$.

2.5. Преобразование уравнения к итерационному виду

Уравнение $F(x) = 0$ преобразуется к виду, пригодному для итерационного процесса, следующим образом:

$$x = x - mF(x),$$

где m — отличная от нуля константа.

В этом случае

$$f(x) = x - mF(x). \quad (2.15)$$

Функция $f(x)$ должна удовлетворять условиям теоремы 2.2. Дифференцируя (2.15), получим

$$f'(x) = 1 - mF'(x). \quad (2.16)$$

Для выполнения условия 3 теоремы 2.2 достаточно подобрать m , так чтобы для всех $x \in [a, b]$

$$|1 - mF'(x)| \leq 1. \quad (2.17)$$

Пример 2.2. Найти решение уравнения $x^4 - 11x^3 + x^2 + x + 0.1 = 0$ методом простой итерации, используя пакет MATLAB.

3. Создайте файл `Func.m` (листинг 2.4), содержащий описание функции $f(x) = x^4 - 11x^3 + x^2 + x + 0.1$.

Листинг 2.4. Файл `Func.m`

```
function z=Func(x)
z=x.^4-11*x.^3+x.^2+x+0.1;
```

4. Создайте файл `Func1.m` (листинг 2.5), содержащий описание функции $f1(x, m, f) := x - m \cdot f(x)$.

Листинг 2.5. Файл `Func1.m`

```
function z=Func1(x,m,f)
z=x-m*feval(f,x);
```

5. Создайте файл `Func2.m` (листинг 2.6), содержащий описание функции $f2$ (2.16).

Листинг 2.6. Файл `Func2.m`

```
function z=Func2(x,m,f)
dx=10^-7;
x1=x+dx;
tmp1=x-m*feval(f,x);
tmp2=x1-m*feval(f,x1);
z=abs((tmp2-tmp1)/dx);
```

6. Постройте графики функций $f1, f2$ (рис. 2.6).

```
>> dx=10^-3;
>> x1=-0.1;x2=0.8;
>> x=x1:dx:x2;
>> m=-0.05;
>> plot(x,Func1(x,m,'Func'));
>> hold on
>> plot(x,Func2(x,m,'Func'),'--');
>> grid on
```

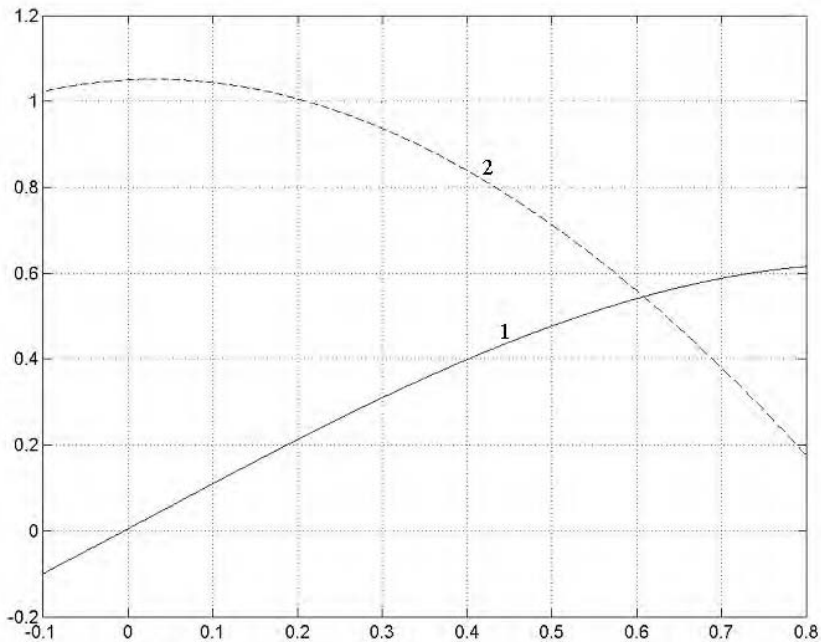


Рис. 2.6. Графики функций $f(x) = x - mF(x) - 1$ и $f'(x) = 1 - mF(x) - 2$

Из рис. 2.6 видно, что на интервале $[0.21; 0.8]$ функция удовлетворяет условиям теоремы 2.2.

- Создайте файл `My_Iter.m` (листинг 2.7), который описывает функцию, возвращающую значение производной на каждом шаге итерационного процесса.

Листинг 2.7. Функция `My_Iter.m`

```
function z=My_Iter(f,x0,eps,q,m)
x(1)=x0;
i=1;
while abs(x(i)-Func1(x(i),m,f))>q/(1-q)*eps
    x(i+1)=Func1(x(i),m,f);
    i=i+1;
end;
z=x;
```

- Задайте параметры итерационного процесса:

```
>> q=0.01;
>> eps=10^-5;
```

9. Вычислите значения корня уравнения на каждом шаге итерационного процесса.

```
>> z=My_Iter('Func',x0,eps,q,m)
```

10. Визуализируйте итерационный процесс (рис. 2.7).

```
>> plot(z,'-o');
```

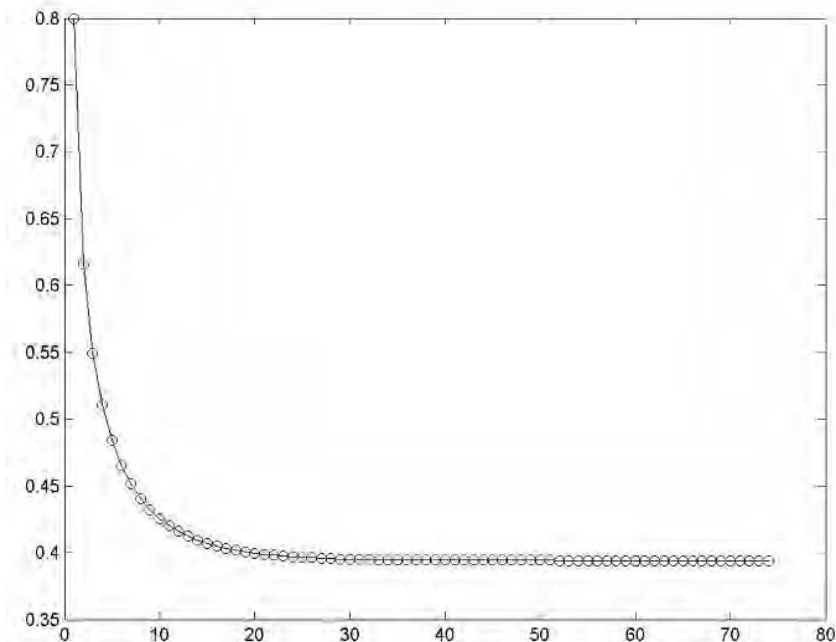


Рис. 2.7. Зависимость значения корня уравнения от номера шага итерационного процесса

11. Выведите точное значение корня.

```
>> Ni=length(z);
```

```
>> z(Ni)
```

```
ans =
```

```
0.3942
```

12. Выведите значения функции.

```
>> Func(z(Ni))
```

```
ans =
```

```
-1.8185e-006
```

Для вычисления нулей функций, зависящих от одной переменной, в пакете MATLAB предусмотрена специальная функция `fzero()`, реализующая

в зависимости от вида функции методы половинного деления, секущих или обратной квадратичной интерполяции. Обращение к данной функции имеет следующий вид:

```
x = fzero(fun,x0,options,P1,P2,...)
[x,fval,exitflag,output] = fzero(...)
```

Здесь:

fun — строковая переменная, содержащая имя файла.

x0 — начальное приближение или интервал поиска решения.

options — параметры, задающие точность и способ представления результатов вычислений.

P1, **P2**, ... — дополнительные аргументы, передаваемые в функцию **fun** (например, **F=feval(FUN,X,P1,P2,...)**).

fval — переменная, в которую функция **fzero()** возвращает значение корня уравнения $f(x) = 0$.

exitflag — переменная, знак которой свидетельствует о наличии корня на данном интервале (например, **exitflag=1** — корень существует);

output — переменная, в которую функция **fzero()** возвращает название метода, использованного для нахождения корня уравнения.

Пример 2.3. Решение уравнения $x^4 - 11x^3 + x^2 + x + 0.1 = 0$ с использованием функции **fzero()**.

```
>> x=fzero('Func',0.8)
x =
    0.3942
```

Поиск корня на отрезке $[-2, 2]$:

```
>> x=fzero('Func',[-2,2])
x =
    0.3942
```

Поиск корня с точностью до 10^{-2} , вывод на экран значение корня и соответствующего значения функции на каждом шаге итерационного процесса:

```
>> x=fzero('Func',0.8,optimset('TolX',10^-2,'disp','iter'))
```

Func-count	x	f(x)	Procedure
1	0.8	-3.6824	initial
2	0.777373	-3.32063	search
3	0.822627	-4.06625	search
4	0.768	-3.17712	search
5	0.832	-4.23184	search
6	0.754745	-2.98039	search
7	0.845255	-4.47271	search