# ИСМО

## 1. Классификация и структура интеллектуальных агентов

Интеллектуальный агент — программа, самостоятельно выполняющая задание, указанное пользователем, в течение длительных промежутков времени.

**Функции**:

* Поиск источников
* Запрос
* Сопоставление и проверка их на соответствие критериям поиска
* Выдача ответа в удобной для пользователя форме. Классификация:

Агенты с **простым поведением** (только на основе текущих знаний).

Агенты с **поведением основанном на модели** (могут оперировать со средой, лишь частично поддающейся наблюдению. Внутри агента хранится представление о той части, что находится вне границ обзора).

**Целенаправленные агенты** (схожи с предыдущим типом, однако они, помимо прочего, хранят информацию о тех ситуациях, которые для них желательны. Это дает агенту способ выбрать среди многих путей тот, что приведет к нужной цели.)

**Практичные агенты** (Целенаправленные агенты различают только состояния, когда цель достигнута, и когда не достигнута. Практичные агенты, помимо этого, способны различать, насколько желанно для них текущее состояние. Такая оценка может быть получена с помощью «функции полезности», которая проецирует множество состояний на множество мер полезности состояний.)

Существует несколько типов агентов:

**Физический** Агент — агент, воспринимающий окружающий мир через некоторые сенсоры и действующий с помощью манипуляторов.

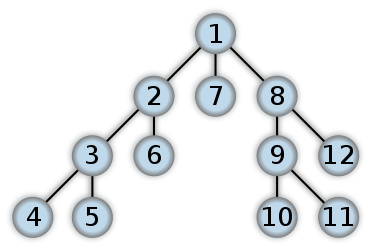
**Временной** агент — агент, использующий изменяющуюся с ходом времени информацию и предлагающий некоторые действия или предоставляющий данные компьютерной программе или человеку, и получающий информацию через программный ввод.

## 2. Поиск в глубину, в ширину, двунаправленный поиск

**Алгоритм поиска в глубину**

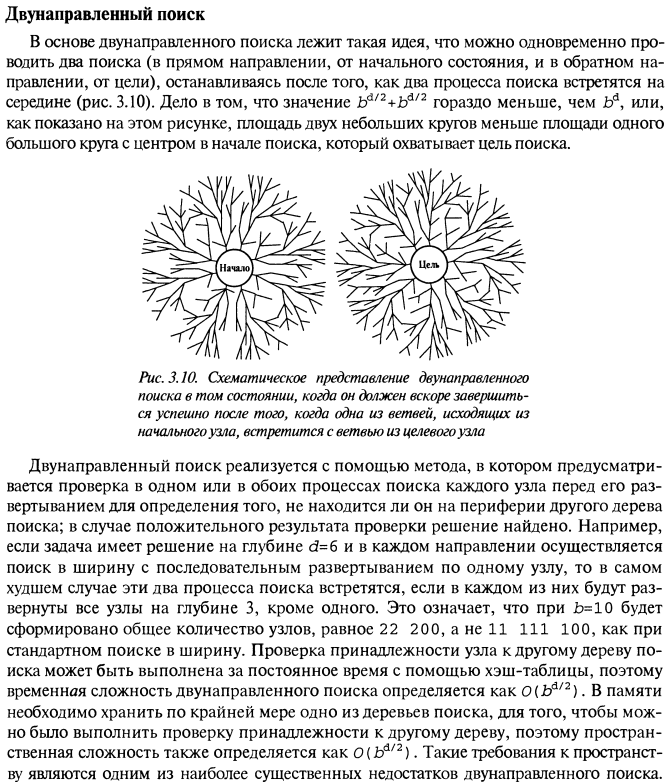
Пусть задан граф , где – множество вершин графа, – множество ребер графа. Предположим, что в начальный момент времени все вершины графа окрашены в белый цвет. Выполним действия:

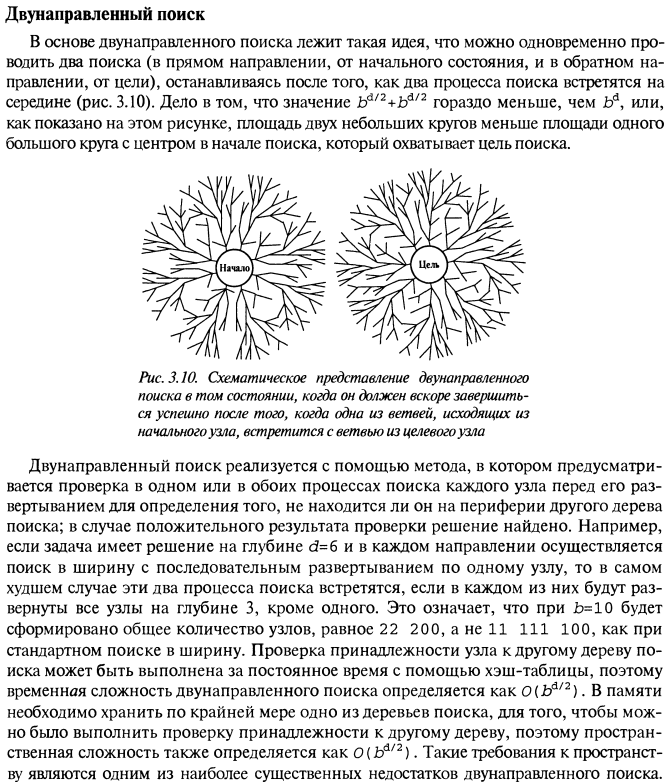
1. Пройдем по всем вершинам . Если вершина белая, то выполняем для нее процедуру DFS(v).
2. Процедура DFS – параметр .
   1. Перекрашиваем вершину в серый цвет.
   2. Для всякой вершины , смежной с вершиной и окрашенной в белый цвет, рекурсивно выполняем процедуру DFS(w).



**Алгоритм поиска в ширину**

1. Поместить узел, с которого начинается поиск, в изначально пустую очередь.
2. Извлечь из начала очереди узел и пометить его как серый.
   1. Если узел является целевым узлом, то завершить поиск с результатом «успех».
   2. Иначе, в конец очереди добавить всех преемников узла , которые белые и не находятся в очереди.
3. Если очередь пуста, то все узлы связного графа были просмотрены, следовательно, целевой узел недостижим из начального. Завершение поиска.
4. Вернуться к п. 2.





## 3. Методы машинного обучения

**Машинное обучение** (англ. *machine learning*, ML) – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

Различают **два типа обучения**:

**Способы** машинного обучения:

Обучение **с учителем** – для каждого прецедента задаётся пара «ситуация, требуемое решение»:

* 1. Искусственная нейронная сеть (Глубокое обучение)

1. Метод коррекции ошибки
2. Метод обратного распространения ошибки
3. Метод опорных векторов

Обучение **без учителя** – для каждого прецедента задаётся только «ситуация», требуется сгруппировать объекты в кластеры, используя данные о попарном сходстве объектов, и/или понизить размерность данных:

Обучение **с подкреплением** – для каждого прецедента имеется пара «ситуация, принятое решение»:

1. Генетический алгоритм.

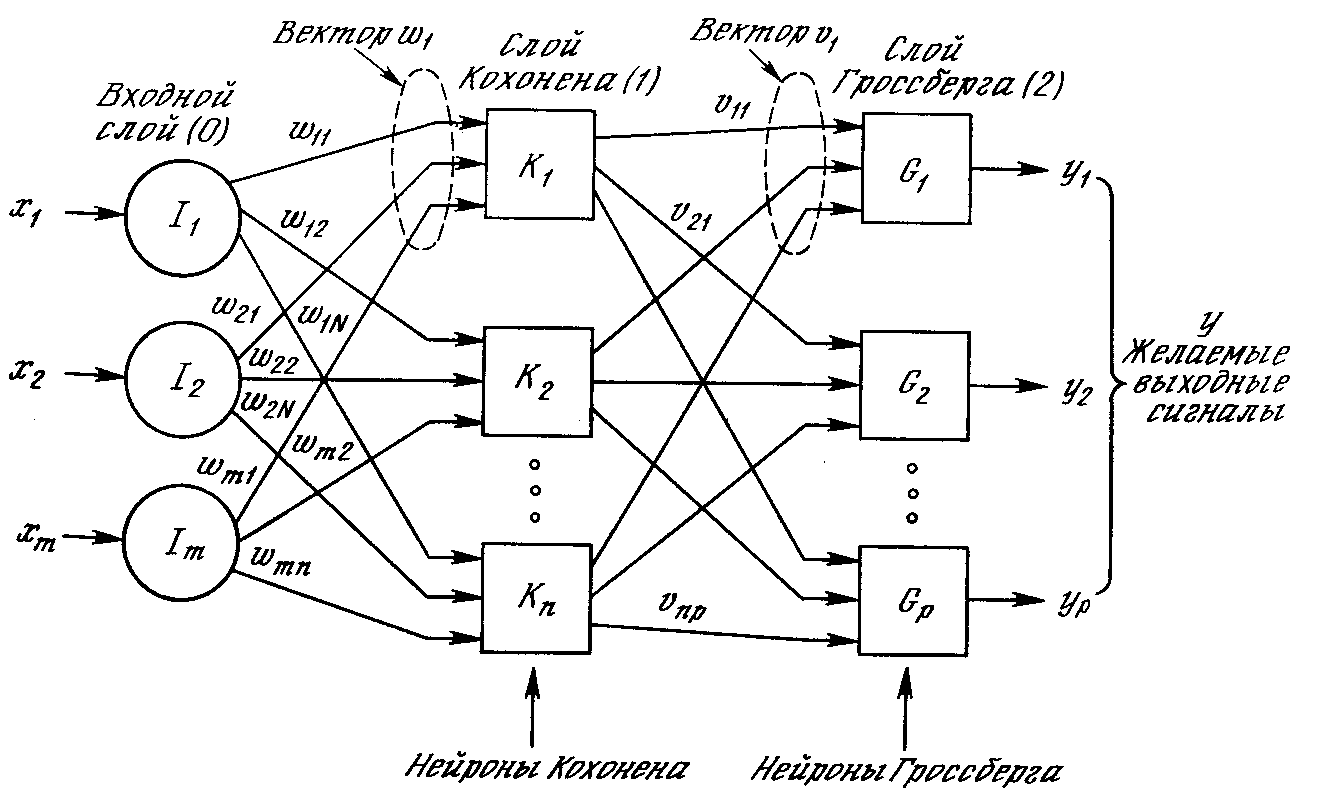
* Байесовская сеть

**Классические задачи**, решаемые с помощью машинного обучения:

* Классификация, как правило, выполняется с помощью обучения с учителем на этапе собственно обучения.
* Кластеризация, как правило, выполняется с помощью обучения без учителя
* Регрессия
* Обнаружение аномалий

## 4. Сети встречного распространения: сети Кохонена и нейроны Гроссберга

Сеть встречного распространения функционирует подобно столу справок, способному к обобщению. В процессе обучения **входные векторы ассоциируются с соответствующими выходными векторами**.



**Рис. 4.1. Сеть с встречным распознаванием без обратных связей**

Слои Кохонена

В своей простейшей форме слой Кохонена функционирует в духе **«победитель забирает все»**, т. е. для данного входного вектора **один и только один нейрон Кохонена выдает на выходе логическую единицу, все остальные выдают ноль**.

Подобно нейронам большинства сетей выход NET каждого нейрона Кохонена является просто суммой взвешенных входов:



где NETj – это выход NET нейрона Кохонена *j*,

Нейрон Кохонена с максимальным значением NET является «победителем». Его выход равен единице, у остальных он равен нулю.

Слой Гроссберга функционирует в сходной манере:

,

Нейроны слоя Гроссберга вычисляют взвешенную сумму своих входов. Функция активации – линейная. Слой Гроссберга дает на выходе линейную комбинацию своих векторов весов, коэффициенты комбинации задаются входами слоя Гроссберга.

## 5. Синхронное обучение Хебба. Двунаправленная ассоциативная память.

Алгоритм обучения Хэбба

В методе Хэбба обучение является исключительно локальным явлением, охватывающим только два нейрона и соединяющий их синапс; не требуется глобальной системы обратной связи для развития нейронных образований.Идея алгоритма выражается следующим равенством:

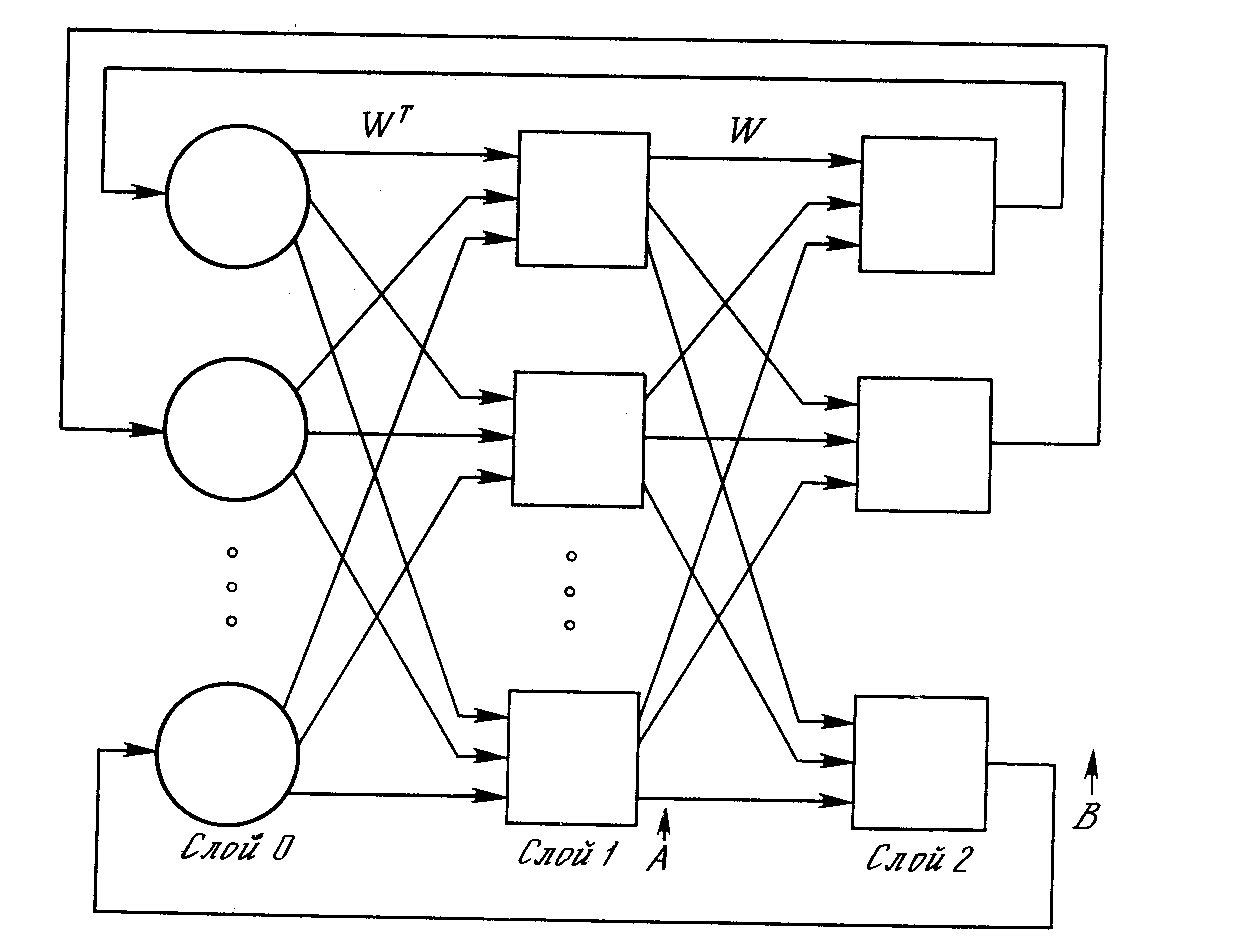
*w*ij(*t*+1) = *w*ij(*t*) + NETi NETj,

где *w*ij(*t*) – сила синапса от нейрона *i* к нейрону *j* в момент времени *t;* NETi – уровень возбуждения предсинаптического нейрона; NETj – уровень возбуждения постсинаптического нейрона.

Двунаправленная ассоциативная память

Двунаправленная ассоциативная память (ДАП) является гетероассоциативной; входной вектор поступает на один набор нейронов, а соответствующий выходной вектор вырабатывается на другом наборе нейронов.

**Выходной нейрон j-го слоя связан с входным нейроном j-го слоя.**



Конфигурация двунаправленной ассоциативной памяти

Этот процесс повторяется до тех пор, пока сеть не достигнет стабильного состояния, в котором ни вектор **А**, ни вектор **В** не изменяются.

Вычисляя сумму взвешенных входов и вычисляя по ней значение функции активации *F,* этот процесс может быть выражен следующим образом:

**В** = *F*(**AW**),

Аналогично

**A** = *F*(**BW**t)

Обычно сеть обучается распознаванию множества образов. Обучение производится с использованием обучающего набора, состоящего из пар векторов A и B.



## 6. Индуктивный алгоритм построения дерева решений ID3

Алгоритм ID3 подобно методу исключения кандидата обеспечивает изучение понятий на примерах. В алгоритме ID3 понятия представляются в виде *дерева решений*.

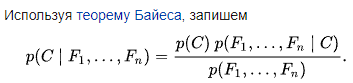
**Алгоритм:**

1. Взять все неиспользованные признаки и посчитать их энтропию относительно тестовых образцов
2. Выбрать признак, для которого **энтропия минимальна** (т.е. неопределенности нет)
3. Сделать узел дерева, содержащий этот признак

С помощью этого дерева можно классифицировать клиента, тип которого неизвестен: для каждого внутреннего узла проверяется значение соответствующего свойства для данного клиента и осуществляется переход по соответствующей ветви. Процесс завершается при достижении конечного узла, определяющего класс объекта.



## 7. Вероятностный вывод в байесовских сетях. Наивные байесовские сети.

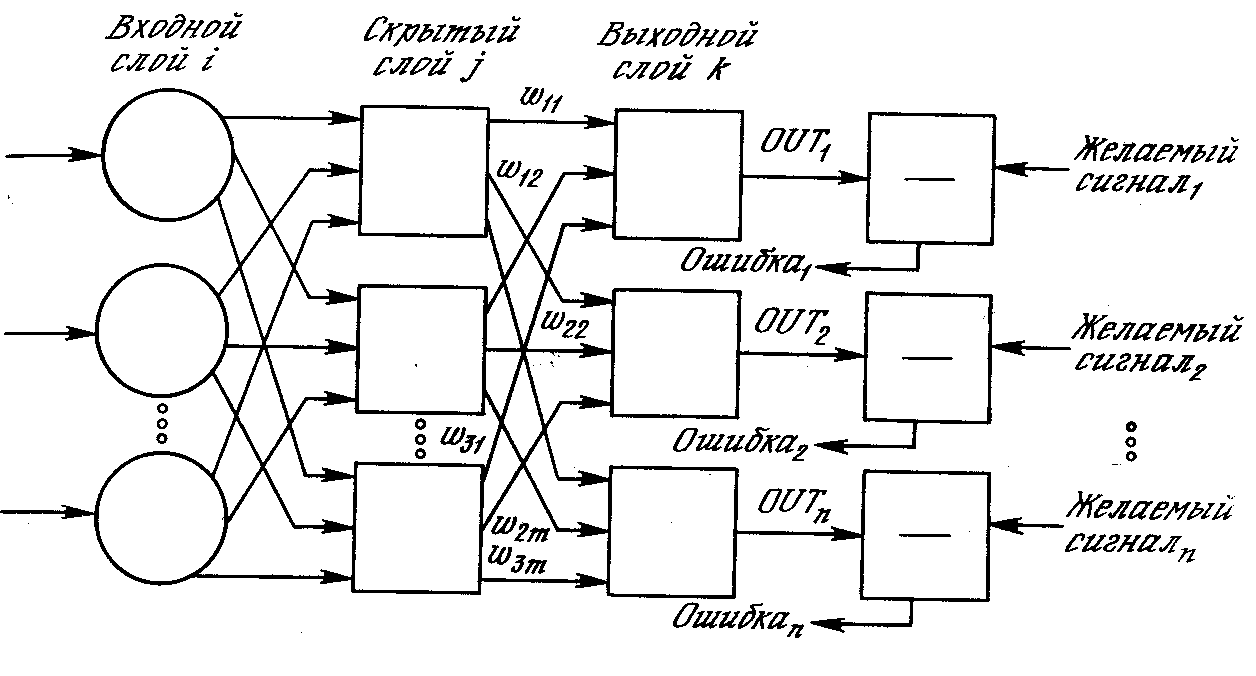


**Наи́вный ба́йесовский классифика́тор** — простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими (наивными) предположениями о независимости.

Упрощенный классификатор:

Все параметры модели могут быть аппроксимированы относительными частотами из набора данных обучения. Это оценки максимального правдоподобия вероятностей. Непрерывные свойства, как правило, оцениваются через нормальное распределение.

## 8. Нейронные сети обратного распространения



Двухслойная сеть обратного распространения

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

## 9. Обобщающая способность моделей машинного обучения

Обобщающая способность — это способность аналитической модели, построенной на основе машинного обучения выдавать правильные результаты не только для примеров, участвовавших в процессе обучения, но и для любых новых, которые не участвовали в нем.

Способность к обобщению модель может приобрести только за счет большого числа разнообразных комбинаций входных и выходных значений в примерах обучающего множества.

Для проверки обобщающей способности модели используется тестовое множество, формируемое из примеров обучающей выборки, не использовавшихся в процессе обучения.

## 10. Нечеткий логический вывод Мамдани и Сугено

Основой для проведения операции нечеткого логического вывода является база правил, содержащая нечеткие высказывания в форме «Если-то» и функции принадлежности для соответствующих лингвистических термов. При этом должны соблюдаться следующие условия:

1. Существует хотя бы одно правило для каждого лингвистического терма выходной переменной.
2. Для любого терма входной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве предпосылки (левая часть правила).

Метод Мамдани

Пусть есть продуктивные правила:

где x,y,z – имена переменных входа и выхода, а – упрощенная запись заданных непрерывных функций принадлежности, при этом четкое значение необходимо определить при текущих переменных и .

1. Введение нечеткости: находятся текущие степени принадлежности для предпосылок правил:

2. Определяем уровни «отсечения»:

3. Находятся «усеченные» функции принадлежности для предпосылок каждого правила при нечеткой композиции:

4. Нечеткая композиция (свертка) производится объединением найденных усеченных функций: с помощью поточечного суммирования

5. Приведение к четкости модифицированным методом центроида

Метод Сугено

Пусть есть продуктивные правила:

где x,y,z – имена переменных входа и выхода, а – упрощенная запись заданных непрерывных функций принадлежности, при этом четкое значение необходимо определить при текущих переменных и .

1. Введение нечеткости: находятся текущие степени принадлежности для предпосылок правил:

2. Определяем уровни «отсечения»:

3. Определяем «индивидуальные выходы»

где коэффициенты и задаются.

4. Методом центроида приводим к четкости переменной вывода:

# ПиРП

## 1.Понятие нити и основные отличия от процесса

**Процесс** - экземпляр программы во время выполнения (независимый объект, которому выделены системные ресурсы), например, процессорное время и память.

**Поток** - это последовательность кода, которая выполняется в рамках процесса.

Главное отличие процессов от потоков, в том, что **процессы изолированы друг от друга**, используют разные адресные пространства, в то время, когда один поток изменяет ресурс процесса, это изменение сразу же становится видно другим потокам этого процесса.

## 2.Ускорение и эффективность вычислений. Закон Амдала

**Ускорение**, для распараллеленного алгоритма определяется величиной:

т.е. как **отношение времени решения задач** **последовательно** **к времени выполнения параллельного алгоритма**.

**Эффективность** использования параллельным алгоритмом определяется:

Величина эффективности определяет **среднюю долю времени** выполнения алгоритма, в течение которой **процессоры реально задействованы** для решения задачи.

**Закон Амдала** показывает ограничение роста производительности с увеличением количества вычислителей. Суммарное время выполнения задачи на параллельной системе не может быть меньше времени выполнения самого медленного фрагмента. Ускорение выполнения программы ограничено временем, необходимым для выполнения её последовательных инструкций.

Ускорение процесса вычислений при использовании *N* процессоров ограничивается величиной:

где  – доля последовательных вычислений.

## 3.Модели параллельно-последовательного программирования MPMD и SPMD

В **SPMD** (single program, multiple data), множество процессоров одновременно выполняют одну и ту же программу из разных её мест (имеется в виду участки кода).

**MPMD** (multiple programs, multiple data) — описывает систему, а) где на одном процессоре машины работает мастер-программа, а на других подчиненная программа, работой которой руководит мастер-программа; б) где на разных узлах машины работают разные программы, которые по-разному обрабатывают один и тот же массив данных, большей частью они работают независимо друг от друга, но время от времени обмениваются данными для перехода к следующему шагу.

## 4.Понятие мьютекса и основные операции с ним. Пример использования мьютекса для синхронизации нитей

**Мьютекс** — это специальный объект для синхронизации потоков. Мьютекс обеспечивает механизм, чтобы доступ к объекту в определенное время был только у одного потока.

Пример:

static Mutex mutexObj = new Mutex(); ### создаем объект мьютекса

static void Main(string[] args)

{

Thread myThread = new Thread(Count);

}

public static void Count()

{

mutexObj.WaitOne(); ### Останавливаем поток(закрываем на замок)

…

mutexObj.ReleaseMutex(); ### Когда всё выполнили, освобождаем поток

}

## 5.Понятие семафора. Пример использования семафора для синхронизации нитей.

Семафоры позволяют ограничить доступ определенным количествам объектов.

new Semaphore(3, 3);

Видим 2 параметра: первый – какому числу объектов изначально будет доступен семафор, а второй параметр указывает, какой максимальное число объектов.

Аналогично мьютексу, методом sem.WaitOne() начинаем ожидание, поток заполняет место, затем освобождаем семафор sem.Release(), затем на это место может идти другой поток.

Например: Есть читатели, которые приходят в библиотеку 3 раза в день. Пусть будет ограничение, что единовременно в библиотеке не может находиться больше 3 читателей.

static Semaphore sem = new Semaphore(3, 3);

int count = 3; ### счетчик чтения

public Reader(int i) {

myThread = new Thread(Read);

…

}

public void Read() {

while (count > 0)

{

sem.WaitOne();

…

sem.Release();

Thread.Sleep(1000);

## 6. Режимы выполнения независимых частей программы: многозадачные, параллельные и распределенные вычисления.

При рассмотрении проблемы организации параллельных вычислений следует различать следующие возможные режимы выполнения независимых частей программы:

* *многозадачный режим (режим разделения времени)*, при котором для выполнения процессов используется единственный процессор; данный режим является псевдопараллельным, все остальные процессы находятся в состоянии ожидания своей очереди на использование процессора;
* *параллельное выполнение*, когда в один и тот же момент времени может выполняться несколько команд обработки данных; данный режим вычислений может быть обеспечен не только при наличии нескольких процессоров, но реализуем и при помощи конвейерных устройств;
* *распределенные вычисления*; данный термин обычно используют для указания параллельной обработки данных, при которой используется несколько обрабатывающих устройств, достаточно удаленных друг от друга.

## 7. Основные понятия и способы реализации технологий OpenMP и MPI и различия между ними

OpenMP - это способ программирования на устройствах с общей памятью. Это означает, что параллелизм возникает там, где каждый параллельный поток имеет доступ ко всем данным.

MPI - это способ программирования на устройствах с распределенной памятью. Это означает, что параллелизм возникает там, где каждый параллельный процесс работает в своем собственном пространстве памяти в изоляции от других.

## 8. Классификация вычислительных систем по систематике Флинна. Архитектуры SMP и MPP

В систематике Флинна различают следующие основные типы систем:

SISD (Single Instruction, Single Data) - системы, в которых существует одиночный поток команд и одиночный поток данных;

SIMD (Single Instruction, Multiple Data) - системы c одиночным потоком команд и множественным потоком данных; каждый момент времени может выполняться одна и та же команда для обработки нескольких информационных элементов;

MISD (Multiple Instruction, Single Data) - системы, в которых существует множественный поток команд и одиночный поток данных; введение подобного класса предпринимается для полноты системы классификации;

MIMD (Multiple Instruction, Multiple Data) - системы c множественным потоком команд и множественным потоком данных; к подобному классу систем относится большинство параллельных многопроцессорных вычислительных систем.

Различают мультипроцессоры по способу построение общей памяти. Использование единой **общей** памяти служит основой для построения симметричных мультипроцессоров (**SMP**), что обеспечивает однородный доступ к памяти. Системы с **распределённой** памятью уже не обеспечивают общий доступ ко всей имеющейся в системах памяти. Данный подход используется при построении массивно-параллельных процессоров (**MPP**)

## 9. Пути достижения параллелизма. Понятие конвейерной реализации обрабатывающих устройств.

Достижение параллелизма возможно только при выполнимости следующих требований к архитектурным принципам построения вычислительной системы:

*независимость функционирования отдельных устройств ЭВМ*

*избыточность элементов вычислительной системы* - организация избыточности может осуществляться в следующих основных формах:

использование специализированных устройств (отдельных процессоров для целочисленной и вещественной арифметики, регистры, кэш);

дублирование устройств ЭВМ, например, нескольких однотипных процессоров или нескольких устройств оперативной памяти.

Конвейерная реализация обрабатывающих устройств - форма при которой выполнение операций представляется в виде исполнения последовательности составляющих операцию подкоманд; как результат, при вычислениях на таких устройствах могут находиться на разных стадиях обработки одновременно несколько различных элементов данных.

## 10. Понятие распределенных вычислений и распределенной системы.

Распределенные вычисления - параллельная обработка данных, при которой используется несколько обрабатывающих устройств, достаточно удаленных друг от друга. Эффективная обработка данных при таком способе организации вычислений возможна только для алгоритмов с низкой интенсивностью потоков межпроцессорных передач данных; перечисленные условия является характерными, например, при организации вычислений в многомашинных вычислительных комплексах.

Распределённая система — система, для которой отношения местоположений элементов играют существенную роль с точки зрения функционирования системы. Для таких систем характерно распределение функций, ресурсов между множеством узлов и отсутствие единого управляющего центра, поэтому выход из строя одного из узлов не приводит к полной остановке всей системы.

# ДМЭ

## Производственные функции. Понятие.Виды.

Производственные функции (ПФ) – это уравнение, устанавливающее связь между факторами [производства](https://sprintinvest.ru/teoriya-i-izderzhki-proizvodstva) (то есть входными [ресурсами](https://sprintinvest.ru/upravlenie-resursami-predpriyatiya-problemy-i-sistemy)) и общим продуктом (то есть выпуском).Q=f(K,L) K – затраты капитала, L – затраты труда, Q– объем выпуска.

Существует три основных типа производственных функций:

**(1)** **линейная** производственная функция**P = aL+ bK**, где**P** —общий продукт,**a** —производительность L единиц труда,**b** — производительность K единиц капитала.,

**(2)** [производственная функция **Кобба-Дугласа**](https://sprintinvest.ru/proizvodstvennaya-funkciya-kobba-duglasa-ponyatie-i-formula-rascheta) **Q = A Ka Lb**, где **Q** — общий продукт,**K** — единицы капитала, **L** — единицы труда,**A** – [общая факторная производительность](https://sprintinvest.ru/sovokupnaya-faktornaya-proizvoditelnost-opredelenie-i-formula-rascheta), **a** и **b** — эластичность выпуска капитала и труда соответственно и

**(3)** производственная функция с фиксированными пропорциями (также называемая производственной функцией **Леонтьева**) **Q = min (aK, bL)**, где **Q** — общий продукт, **a** и **b** — соответственно коэффициент производства капитала и труда, **K** и **L** — соответственно единицы капитала и труда.

## Формулировка модели Рамсея-Касса-Купманса.

Модель Рамсея — Касса — Купманса (модель Рамсея) — [неоклассическая](https://wiki2.info/%D0%9D%D0%B5%D0%BE%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%8D%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F) модель равновесного [экономического роста](https://wiki2.info/%D0%AD%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%80%D0%BE%D1%81%D1%82), в которой «траектория» [потребления](https://wiki2.info/%D0%9F%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) и [сбережений](https://wiki2.info/%D0%A1%D0%B1%D0%B5%D1%80%D0%B5%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F) определяются на основе решения задачи [оптимизации](https://wiki2.info/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) домашних хозяйств и фирм в условиях [совершенной конкуренции](https://wiki2.info/%D0%A1%D0%BE%D0%B2%D0%B5%D1%80%D1%88%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BD%D0%BA%D1%83%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F).

В модели рассматривается [закрытая экономика](https://wiki2.org/ru/%D0%97%D0%B0%D0%BA%D1%80%D1%8B%D1%82%D0%B0%D1%8F_%D1%8D%D0%BA%D0%BE%D0%BD%D0%BE%D0%BC%D0%B8%D0%BA%D0%B0). Предпосылка о закрытой экономике означает, что произведенный продукт тратится на инвестиции и потребление, экспорт и импорт отсутствуют, сбережения равны инвестициям: , , амортизация, коэффициент амортизации, I - инвестиции

## Фазовые переменные в моделях экономической динамики. Содержательный смысл.

Те **переменные**, для которых выписываются уравнения, называются зависимыми или **фазовыми**.

## Переменные управления в моделях экономической динамики. Суть и ограничения, накладываемые на них.

**Переменные управления** – это независимые переменные процесса. Это собственно характеристики управляющей системы. К переменным управления относят: организационную структуру, процессы управления, руководство (лидерство) и поведение в организации.

## Условия принципа максимума Понтрягина.

**Формулировка принципа максимума**. Оптимальным является управление из области допустимых значений, которое обеспечивает максимум гамильтониана.

Согласно принципу максимума Понтрягина величина оптимального управления равна величине управления на одном из концов допустимого диапазона.

Запишем условия применения принципа максимума:

, .

где  – двойственные к фазовым переменные;  – фазовые переменные;  – гамильтониан.

## Понятие оптимальной траектории развития экономической системы.

Это кривая изменения (фазового) состояния экономики во времени под воздействием различных управляемых факторов

Задача экономического управления (планирования) заключается в том, чтобы была реализована наиболее эффективная (оптимальная) траектория.

Оптимальной называется траектория, обеспечивающая на протяжении изучаемого периода лучшие результаты развития системы (лучшие — относительно заданного критерия оптимальности)

## Понятие квазистационарной траектории развития экономической системы

Происходит относительно медленное изменение *условий*, вследствие чего возможно при принятии долгосрочных хозяйственных *решений* опираться на текущие *показатели*

## Что из себя представляет информационный паспорт модели экономической динамики.

Исходная информация задачи – информационный паспорт.

Информационный паспорт для стационарной модели имеет вид:

## Релейное переключение в стратегии оптимального управления. Понятие.

Оптимальное по быстродействию управление есть кусочно-постоянные или релейные функции времени.

## Стационарность и квазистационарность в моделях экономической динамики.

Экономика достигает стационарного состояния, когда чистое накопление капитала на единицу эффективного труда прекращается.

Квазистационарным будем называть процесс, если его параметры меняются существенно медленнее остальных динамических процессов, протекающих в системе

# МИСАСД

## 1.Платформа обработки статистических данных Apache Spark.

Apache Spark – это Big Data фреймворк с открытым исходным кодом для распределённой пакетной и потоковой обработки неструктурированных и слабоструктурированных данных, входящий в экосистему проектов Hadoop.

Apache Spark состоит из следующих компонентов:

* Ядро (Core);
* SQL – инструмент для аналитической обработки данных с помощью SQL-запросов;
* Streaming – надстройка для обработки потоковых данных;
* MLlib – набор библиотек машинного обучения;
* GraphX – модуль распределённой обработки графов.

## 2.Платформа распределенных вычислений Apache Hadoop.

**Hadoop** – это программная **платформа** (software framework), **позволяющая выполнять распределенную обработку огромных массивов данных**. При этом Hadoop обеспечивает надежность, эффективность и масштабируемость.

Платформа Hadoop состоит из нескольких элементов. В основании лежит распределенная файловая система Hadoop Distributed File System (**HDFS**), распределяющая файлы по нескольким узлам хранения в кластере Hadoop. Над файловой системой HDFS (в рамках рассмотрения этой статьи) располагается механизм MapReduce, состоящий из узлов типов JobTracker и TaskTracker.

## 3. Линейный и нелинейный регрессионный анализ данных.

Регрессионный анализ (regression analysis) – это метод изучения статистической взаимосвязи между одной зависимой количественной зависимой переменной от одной или нескольких независимых количественных переменных.

Уравнение регрессии – математическая функция, которая подбирается на основе исходных статистических данных зависимой и объясняющих переменных.

Функции, нелинейные по объясняющим переменным, можно свести к линейным с помощью замены переменных. Функции, нелинейные по оцениваемым параметрам и переменным-факторам, сводят к линейным моделям с помощью логарифмирования и замены переменных. В случае невозможности подбора линеаризующего преобразования для оценки параметров используют методы нелинейной оптимизации на основе исходных данных.

Библиотека python: sklearn Scikit-learn

## 4. Дискриминантный анализ

Линейный дискриминантный анализ основан на простой вероятностной модели, которая вычисляет условную вероятность принадлежности классифицируемого объекта X к классу К. Результат предсказания может быть получен по формуле Байеса. В качестве результата предсказания выбирается класс, для которого условная вероятность максимальна.

## 5.Деревья решений

Деревья решений и их ансамбли - популярные методы для задач классификации и регрессии машинного обучения. Деревья решений широко используются, поскольку они просты для интерпретации, обрабатывают категориальные особенности, расширяются до настройки мультиклассовой классификации, не требуют масштабирования признаков и способны улавливать нелинейности и взаимодействия функций. Алгоритмы ансамбля деревьев, такие как случайные леса и бустинг, являются одними из лучших для задач классификации и регрессии.

Разбиение дерева на ветви происходит так, чтобы энтропия ветви была равна нулю. (Энтропия — это мера случайности или неопределенности.)

## 6. Логистический регрессионный анализ

Логистическая регрессия — это алгоритм классификации машинного обучения, используемый для прогнозирования вероятности категориальной зависимой переменной.

## 7. Иерархический кластерный анализ.

Иерархическая кластеризация – совокупность алгоритмов упорядочивания данных, направленных на создание иерархии (дерева) вложенных кластеров.

## 8. Итерационные методы кластерного анализа. Метод k-средних.

Алгоритм k-средних кластеризует данные, пытаясь **разделить** выборки на **n групп с равной дисперсией**, **минимизируя** **критерий**, известный как инерция или **сумма квадратов внутри кластера**. Этот алгоритм требует указания количества кластеров. Он хорошо масштабируется для большого количества образцов и используется в широком диапазоне областей применения во многих различных областях.

## 9. Метод главных компонент.

PCA (Principal Component Analysis, Метод главных компонент) — это статистическая **процедура уменьшения размерности**. С математической точки зрения **в основе PCA** лежит **нахождение линейно некоррелированных переменных**, называемых главными компонентами. PCA очень часто используется в статистике и машинном обучении (Machine Learning), поскольку **помогает выделить наиболее значимую информацию** из всего датасета, который может состоять из сотни признаков.

## 10. Методы поиска ассоциаций в наборах данных. Метод FP-Growth

В основе алгоритма лежит предобработка БД транзакций, в процессе которой она трансформируется в компактную древовидную структуру, называемую Frequent-Pattern Tree – дерево популярных предметных наборов (откуда и название алгоритма). В дальнейшем для краткости будем называть эту структуру FP-дерево. К **основным преимуществам** данного метода относятся:

1. Сжатие БД транзакций в компактную структуру, которая обеспечивает наиболее эффективное и полное извлечение частых предметных наборов;
2. При построении FP-дерева используется технология «разделяй и властвуй» (англ.: divide and conquer), которая позволяет выполнить декомпозицию одной сложной задачи на множество более простых;
3. Позволяет избежать затратной процедуры генерации кандидатов, характерной для алгоритма Apriori.

# «Компьютерная алгебра»

## Проблема представления целых и действительных чисел в компьютерной алгебре.

Известно, что компьютер – машина с конечной памятью, при этом максимальное целое число, которое можно разместить составляет .

Для хранения чисел произвольной величины используются специальные типы (BigInteger, представляющий собой массив данных).

можно представить в виде массива с множителями

Проблема представления действительных чисел заключается в ограниченной точности, так числа одинарной точности **float** ограничены в точности до 8 знаков после запятой, числа с двойной точностью **double** ограничены 16 знаками.

## Представление элементов кольца полиномов в компьютерной алгебре.

Существует несколько способов представить полином от одной переменной степени , например, с помощью упорядоченного списка его ненулевых коэффициентов. Каждый коэффициент при этом может быть представлен в виде массива как и в случае с целыми числами произвольной длины.

## Алгебраические числа.

Алгебраическим числом называется число **α**, являющееся корнем многочлена от одной переменной с целыми коэффициентами. Если старший коэффициент этого многочлена равен 1, то алгебраическое число называется целым.

## Обобщенный алгоритм Евклида вычисления наибольшего общего делителя в кольце полиномов.

Алгоритм Евклида позволяет найти наибольший общий делитель двух многочленов, т.е. многочлен наибольшей степени, на который делятся без остатка оба данных многочлена.

**Алгоритм состоит из последовательного деления с остатком сначала первого данного многочлена**, *f*(*x*), **на второй**, *g*(*x*):

*f*(*x*) = *g*(*x*)∙*q*1(*x*) + *r*1(*x*), (1)

затем, если *r*1(*x*) ≠ 0, – **второго данного многочлена**, *g*(*x*), **на первый остаток** – на многочлен *r*1(*x*):

*g*(*x*) = *r*1(*x*)∙*q*2(*x*) + *r*2(*x*), (2)

далее, если *r*2(*x*) ≠ 0, – первого остатка, *r*1(*x*), на второй остаток, *r*2(*x*):

*r*1(*x*) = *r*2(*x*)∙*q*3(*x*) + *r*3(*x*), (3)

затем, если *r*3(*x*) ≠ 0, – второго остатка на третий и т.д.

Тогда последний не равный нулю остаток *rn* и будет наибольшим общим делителем исходной пары многочленов *f*(*x*) и *g*(*x*).

## Алгоритм Аткина определения простых чисел.

1. Числа 2, 3 и 5 рассматриваются, как заведомо простые.
2. Создаётся «решето» — список, сопоставляющий каждому натуральному числу *n* из диапазона (5; *limit*] флаг «простое\составное». Изначально все флаги устанавливаются в положение «составное».
3. Для очередного *n* из диапазона (5; *limit*] находится остаток от деления на 60:
   * 1, 13, 17, 29, 37, 41, 49, 53: уравнение — *4x² + y² = n*; (1)
   * 7, 19, 31, 43: уравнение — *3x² + y² = n*; (2)
   * 11, 23, 47, 59: уравнение — *3x² − y² = n* (при *x > y*) (3)
4. Значение остатка, не равное ни одному из вышеуказанных, свидетельствует о том, что число *n* — составное и таким образом исключается из дальнейшего рассмотрения. Иначе, если соответствующее уравнение имеет нечётное число решений (пар натуральных значений *x* и *y*), *n* помечается в «решете» как «простое».
5. Процедура повторяется начиная с шага № 3, пока не будут перебраны все *n ≤ limit*.
6. В заключение для каждого найденного простого *n* все значения, кратные его квадрату, помечаются в «решете» как «составные».

## Алгоритм Ферма разложения целого числа на простые множители.

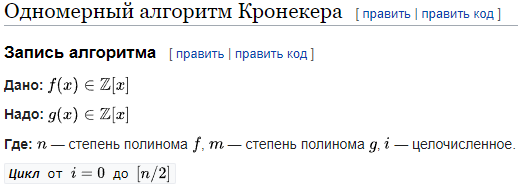
* Вход: Натуральное нечетное число *.*
* Выход: Натуральные делители.

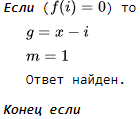
1. Вычислить наименьшее целое число
2. Если , то перезапустить алгоритм со значением .
3. В цикле по вычислять:

Если – целое, то найдено разложение на множители: и

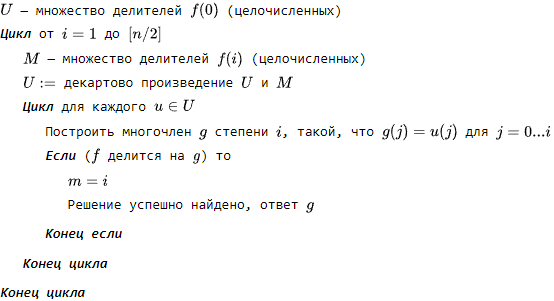
Если один множитель равен 1, то добавить второй множитель к ответу, иначе запустить алгоритм для обоих множителей (Шаг 1).

## Алгоритм Кронекера факторизации полинома.

****

****

****

****

****

****

## Обратная польская нотация. Инфиксная и постфиксная запись выражений.

**Обра́тная по́льская запись –** форма записи математических и логических выражений, в которой операнды расположены перед знаками операций. Также именуется как обратная польская запись, обратная бесскобочная запись, постфиксная нотация. Инфиксное выражение Математические нотации,  Префиксная , Инфиксная, Постфиксная ,Обратная польская запись, польская нотация (запись ) может быть записано так: **1 2 + 4 × 3 +**

## Формальное дифференцирование.

**Формальное дифференцирование** — операция над элементами [кольца многочленов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D1%8C%D1%86%D0%BE_%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D1%87%D0%BB%D0%B5%D0%BD%D0%BE%D0%B2) или кольцом [формальных степенных рядов](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%BD%D0%BE%D0%B9_%D1%80%D1%8F%D0%B4), повторяющая форму производных из [математического анализа](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7).

Определение формального дифференцирования таково: зафиксируем кольцо *R* (не обязательно коммутативное), пусть *A* = *R*[*x*] является кольцом многочленов над *R*. Тогда формальное дифференцирование представляет собой действие над элементами *A*, при котором если



то формальная производная равна



как и в случае многочленов над вещественными или комплексными числами.

## Алгоритмы интегрирования многочленов и рациональных функций.

* **Шаг 1.** Определить вид многочлена в знаменателе дроби (он может иметь действительные, кратные действительные, комплексные и кратные комплексные корни) и в зависимости от вида **разложить дробь на простые дроби**, в числителях которых - неопределённые коэффициенты, число которых равно степени знаменателя.
* **Шаг 2. Определить значения неопределённых коэффициентов**. Для этого потребуется решить систему уравнений, сводящуюся к системе линейных уравнений.
* **Шаг 3. Найти интеграл исходной рациональной функции (дроби) как сумму интегралов полученных простых дробей**, к которым применяются табличные интегралы.

# ППОвМСС

1 **Основные модели механики жидкостей и газов (идеальная среда, несжимаемая жидкость, совершенный газ, ньютоновская жидкость).**

***Жидкостью*** называется, вещество, которое обладает свойством текучести. ***Текучесть*** – свойство жидкости непрерывно и сколь угодно сильно деформироваться под действием минимального срезывающего напряжения.

**Идеальная (совершенная) жидкость – условная жидкость,** которая считается абсолютно несжимаемой, невязкой и не имеющей молекулярного строения.

**Несжимаемая жидкость –** среда, плотность которой постоянна.

Совершенный газ – модель газа, молекулы которого представляются в виде материальных точек, взаимодействующих только при соударениях. Совершенный газ имеет постоянные теплоемкости Сp и Сv, показатель изоэнтропы k= Сp / Сv и молекулярную массу m

**2 Система уравнений Навье–Стокса.**

**Уравнения Навье-Стокса** – это [система дифференциальных уравнений](http://ru.solverbook.com/spravochnik/differencialnye-uravneniya/sistemy-differencialnyx-uravnenij/), которая описывает движение вязкой ньютоновской жидкости либо газа. Так называемая «вязкость» жидкости – это её способность оказывать сопротивление, если какую-то её часть попытаться сдвинуть относительно соседнего слоя (например, при гребле). При этом в жидкости происходит внутреннее [трение](http://ru.solverbook.com/spravochnik/mexanika/dinamika/sily-treniya/).

Система уравнений движения сплошной среды Навье-Стокса

Дополнительные соотношения:

, , , ,

, .

Здесь – [плотность](http://ru.solverbook.com/spravochnik/mexanika/dinamika/massa-plotnost/) жидкости, t – время, р – [давление](http://ru.solverbook.com/spravochnik/mexanika/gidrostatika/davlenie/), – проекции скорости (вектора) на координатные оси, µ– коэффициент динамической вязкости;

Первое уравнение – это уравнение неразрывности. Его физический смысл – это сохранение [массы](http://ru.solverbook.com/spravochnik/mexanika/dinamika/massa-plotnost/) для потока жидкости.

Второе уравнение – это уравнение движения частиц среды.

Третье уравнение – это уравнение состояния.

**3 Граничные условия при решении задач гидродинамики и теплообмена.**

3.1 Граничные условия для скорости на твердой непроницаемой стенке:

1. – граничное условие прилипания (отсутствие тангенциальной составляющей скорости на границе с телом);
2. – граничное условие не протекания (отсутствие нормальной составляющей скорости на границе с телом);
3. Втекающий сверхзвуковой поток;
4. Вытекающий сверхзвуковой поток;
5. Втекающий дозвуковой поток;
6. Вытекающий дозвуковой поток;

3.2 Граничные условия для температуры:

1. – граничное условие I рода (задана температура).
2. – граничное условие II рода (тепловой поток через границу).
3. – граничное условие III рода (теплообмен между телом и средой).

**4. Критерии гидродинамического подобия, их физический смысл**

**Теория подобия** – учение об условиях подобия физических явлений (опирается на размерность величин). Предметом Т.п. является изучение с помощью этих критериев свойств самих явлений.

Характерные масштабы:

*T* – характерный масштаб времени (период колебаний);

*L* – характерный масштаб длины (диаметр);

*U*0 – характерный масштаб скорости (скорость набегающего потока);

*F* – характерный масштаб объемных сил (ускорение свободного падения).

– число Струхаля, характеризует нестационарность процесса; (<< 1 для стационарного процесса)

– число Эйлера, характеризует отношение сил давления к силам инерции;

– число Рейнольдса, характеризует отношение сил инерции к силам вязкости;  – кинематическая вязкость; (до 2000 – ламинарное течение)

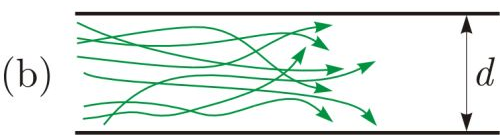
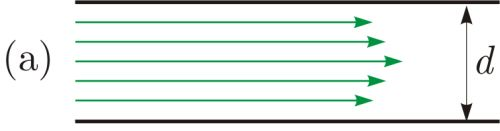
– число Фруда, характеризует отношение сил инерции к силам тяжести.

**5. Понятие турбулентности. Режимы течения**

Первый вид движения, при котором частицы следуют по отчетливо видимым траекториям, представляющим плавные, лишь слегка изменяющиеся со временем, кривые, называются *ламинарным*.

Более распространен второй вид движения с хаотически переплетенными и быстро изменяющимися во времени траекториями, с поперечными и, даже, попятными по отношению к общему движению жидкости перемещениями отдельных малых объемов. Такое нерегулярное, имеющее в малых своих частях случайный характер движение называется *турбулентным*.

С возрастанием скорости ламинарное движение *теряет свою устойчивость*; при этом любые случайные малые возмущения, которые вначале вызывали лишь малые колебания вокруг устойчивого ламинарного движения, начинают быстро развиваться и приводят к новой форме движения жидкости – к турбулентному движению.



**6. Осреднение по Рейнольдсу и Фавру. Тензор турбулентных напряжений**

На основании большого числа систематических наблюдений было отмечено, что существует некоторое характерное для режима движения *критическое* число в последствии названное *критическим числом Рейнольдса*, служащее основным критерием перехода ламинарного движения в турбулентное.



*Uср* – средняя скорость движения в трубе,

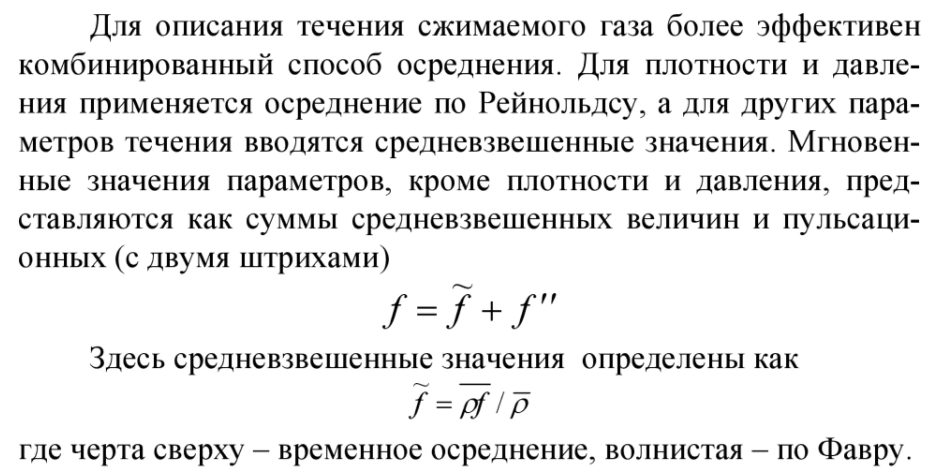
*d* – диаметр трубы,

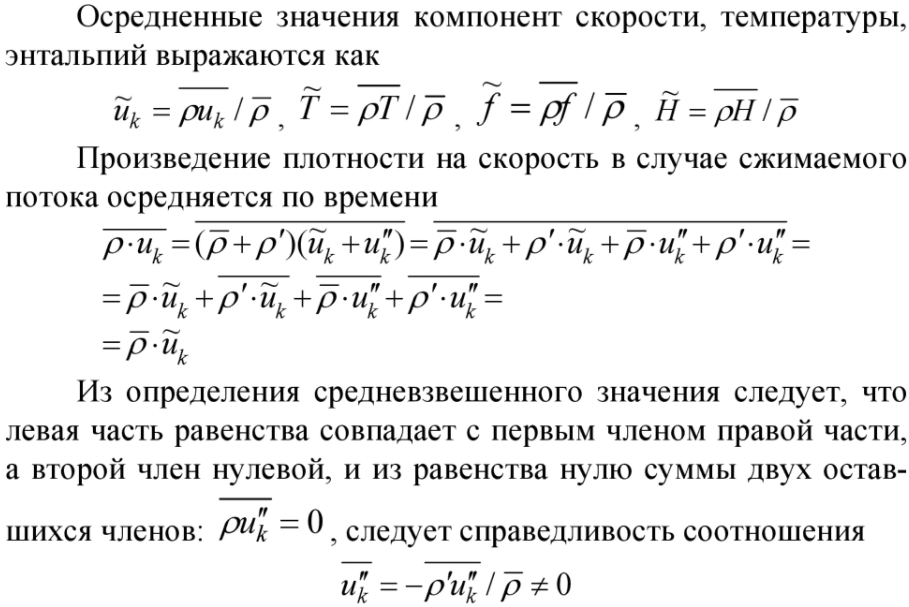
ν - коэффициент кинематической вязкости.

Для исследования турбулентных движений приходится применять особые приемы, связанные с заменой действительного движения некоторой упрощенной схемой *осредненного в пространстве и времени* движения.

Предположим, все параметры колеблются вокруг среднего устойчивого течения. Метод осреднения Рейнольдса заключается в замене случайно изменяющихся характеристик потока (скорость, давление, плотность) суммами осреднённых и пульсационных составляющих.

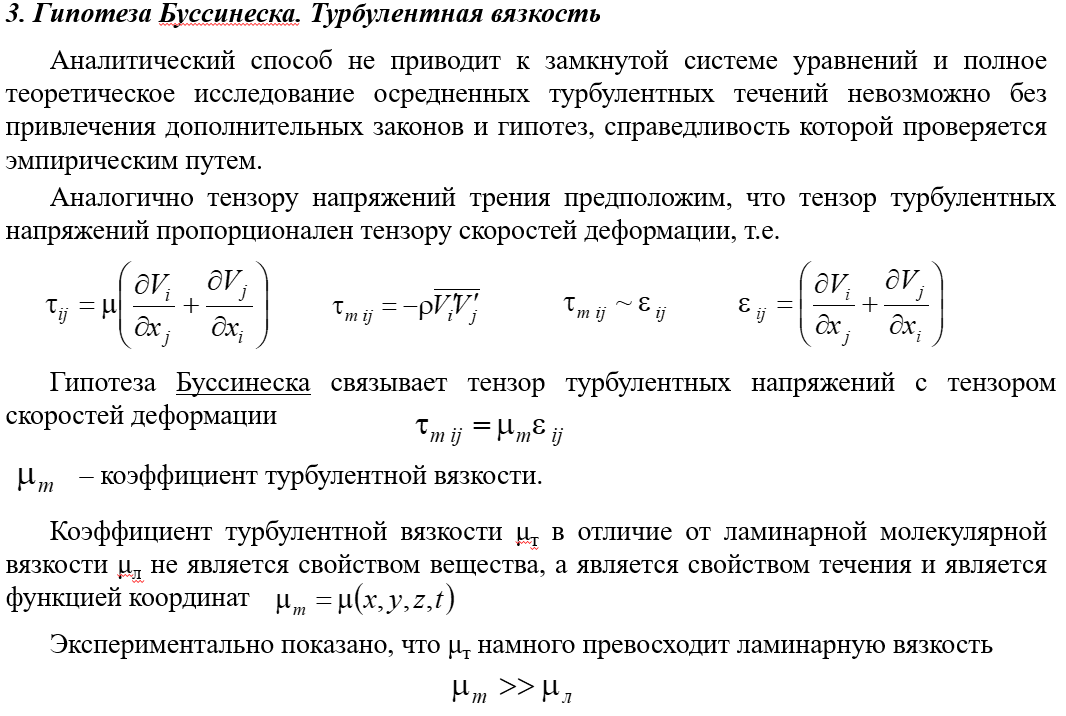
**Осреднение по Фавру (средневзвешенное осреднение):**



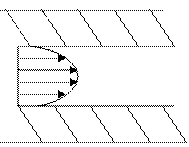


**7.** **Гипотезы турбулентности: Буссинеска, Прандтля, Кармана.**

**Гипотеза Буссинеска.** Тензор турбулентных напряжений пропорционален тензору скоростей деформаций (через коэффициент турбулентной вязкости).

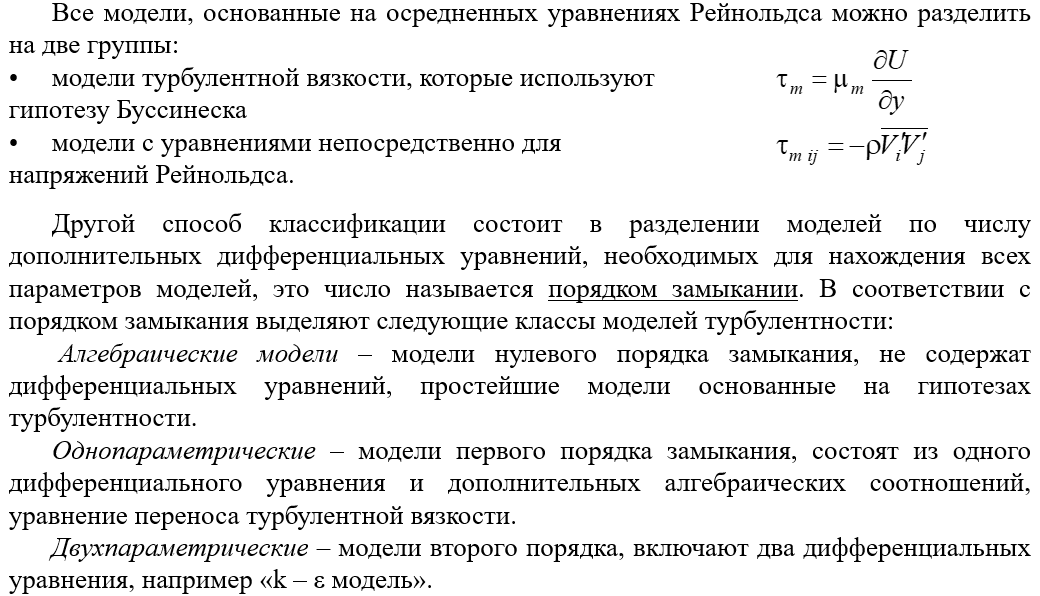


**Гипотеза Прандтля** о пути смешения. Турбулентный поток состоит из турбулентного ядра (в центре трубы) и вязкого ламинарного подслоя. Длина пути смешения – расстояние на котором частица сохраняет свой собственный импульс.

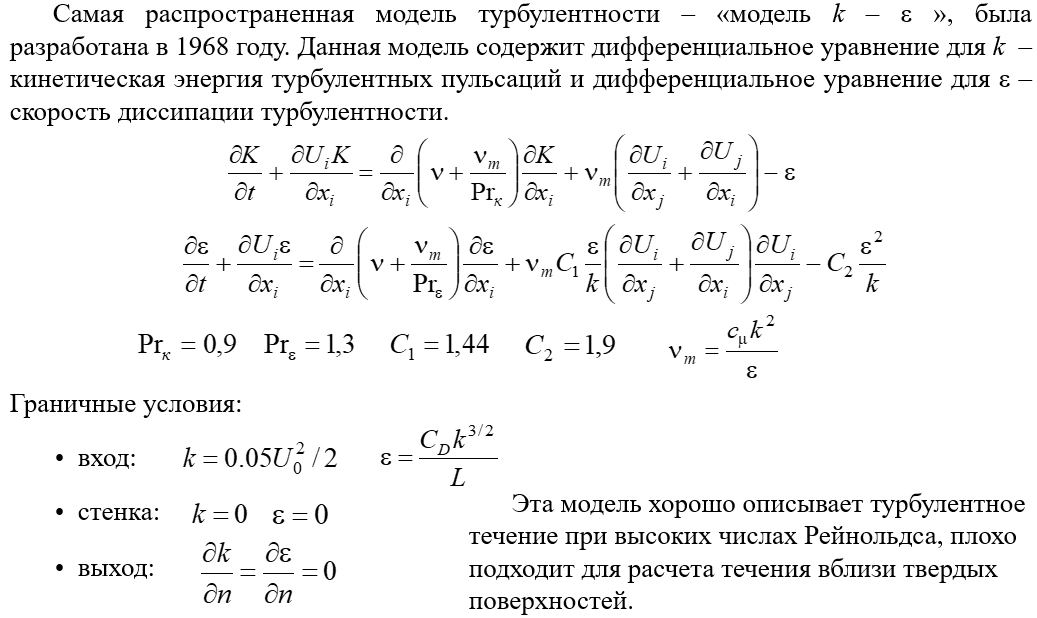


**Гипотеза Кармана (гипотеза подобия Кармана).** В турбулентном течении длина пути смешения пропорциональна расстоянию от стенки. , y – расстояние от стенки, *l* – длина пути смешения.

**8. Классификация моделей турбулентности. Модель турбулентности k-ε.**



Модель турбулентности K-epsilon (k-ε) является наиболее распространенной моделью, используемой в вычислительной гидродинамике (CFD) для **моделирования характеристик среднего потока для условий турбулентного потока**. Первая переносимая переменная - это **турбулентная кинетическая энергия** (k). Вторая переносимая переменная - это **скорость диссипации турбулентной кинетической энергии** (ε).



**10. Модели теории упругости. Закон Гука. Константы упругости материала.**

**Зако́н Гу́ка** — утверждение, согласно которому, [деформация](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), возникающая в упругом теле, пропорциональна приложенной к этому телу [силе](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BB%D0%B0).

**

Рис. 10. Связь между напряжениями и деформациями

Установленная линейная зависимость называется законом Гука и имеет следующий вид:

|  |  |
| --- | --- |
|  | (20) |

где  – модуль упругости материала (модуль Юнга).

Можно показать, что линейная зависимость существует также между касательным напряжением и угловой деформацией

|  |  |
| --- | --- |
| , | (21) |

 - модуль сдвига материала (или модуль упругости второго рода).

Позже С. Пуассон показал, что линейные деформации в точке тела связаны друг с другом следующими соотношениями:

|  |  |
| --- | --- |
|  | (22) |

Константы упругости материала  определяются экспериментально и связаны друг с другом следующим соотношением (23) (для однородного изотропного материала)

|  |  |
| --- | --- |
|  | (23) |

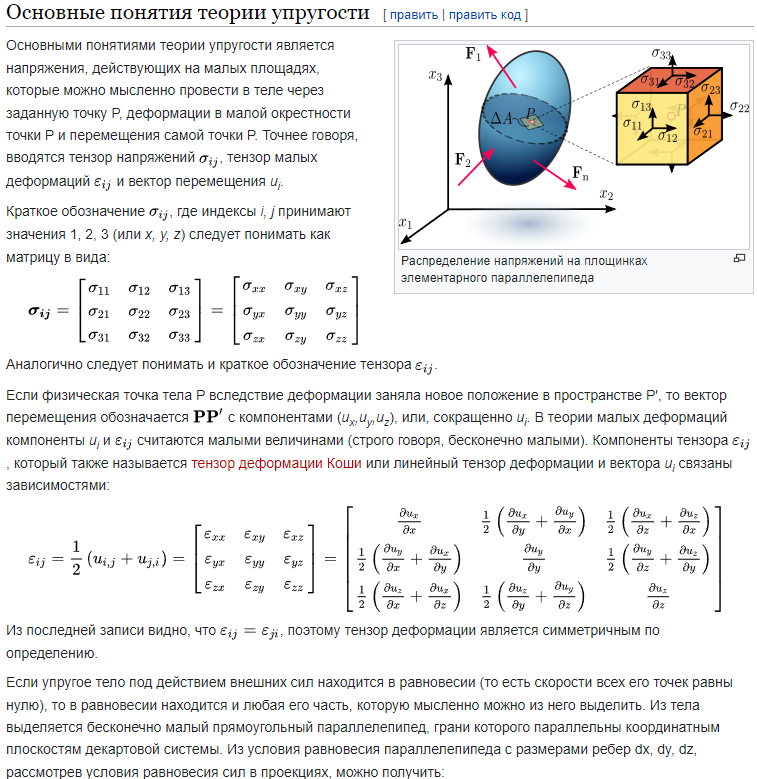
Закон Гука выполняется только при малых деформациях. При превышении [предела пропорциональности](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%B5%D0%BB_%D0%BF%D1%80%D0%BE%D0%BF%D0%BE%D1%80%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8) связь между силой и деформацией становится нелинейной. Для многих сред закон Гука неприменим даже при малых деформациях.

**Модуль Юнга** – физическая величина, характеризующая способность материала сопротивляться упругой деформации.

**Коэффициент Пуассона** – величина отношения относительного поперечного сжатия к относительному продольному растяжению.

**Модуль сдвига** – физ. Величина, характеризующая способность тела сопротивляться сдвиговой деформации.

**11. Тензор напряжений, уравнения равновесия.**

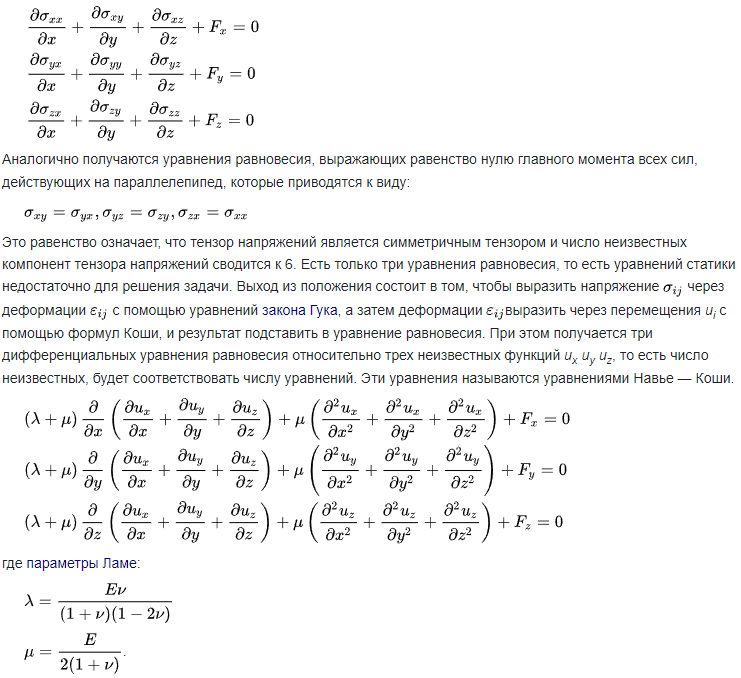


**Тензор напряжений** – совокупность напряжений, действующих на малой площади, которые мысленно можно провести через заданную точку.

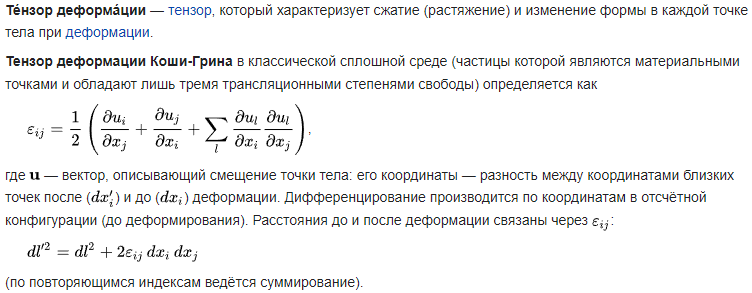
**Уравнения равновесия** представляют собой систему из 3-х дифференциальных уравнений (3-х мерный случай), относительно напряжений, возникающих в теле под действием внешних сил. (может быть записана через напряжения или перемещения)

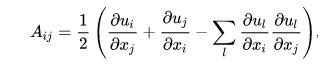
Если упругое тело под действием внешних сил находится в равновесии (скорости всех его частей равны нулю), то и в равновесии находится любая его малая часть.

Уравнения равновесия



**12. Тензор деформаций, уравнения совместности (неразрывности) деформаций.**



Тензор деформаций Альманзи 

Либо тензор малых деформаций

**Уравнения совместности (неразрывности) деформаций**

Деформируемое тело до момента его разрушения остаётся сплошным телом (отсутствие трещин, пор и других разрывов). Отсюда следует, что **перемещения и деформации являются непрерывными функциями координат точки тела**.

|  |  |
| --- | --- |
| , , | (17) |

Продифференцировав выражение (17) один раз по , а затем по , можно получить:



Из этого выражения следует, что если в точке тела известны две линейных деформации в какой – то координатной плоскости, то деформация сдвига в этой плоскости произвольной быть не может.

Таким образом, в произвольной точке деформируемого сплошного тела должны выполняться следующие соотношения между компонентами тензора деформации:

|  |  |
| --- | --- |
| , , , | (32-34) |
|  | (35) |
|  | (36) |
|  | (37) |

Уравнения (32) – (37) называются уравнениями совместности деформаций (уравнениями неразрывности).