

жет уже рассматриваться как сильноионизованный, а его проводимость определяется по ф-ле (9а).

При достаточно редких столкновениях анализ П. п. требует учёта инерции электронов и кинетич. эффектов, таких, как убегание электронов Будкера — Дрейсера (см. *Убегание электроны, пристеночная проводимость, аномальное сопротивление*, а также проводимость за счёт неоклассич. переноса (см. *Переноса процессы*).

Благодаря различию скоростей ионной и электронной компонент, приводящему к эффекту Холла, траектории ионов и электронов в плазменных объёмах могут иметь совершенно разный вид (рис. 2). Так, напр., в осесимметричных плазменных ускорителях с замкнутым дрейфом ионы идут вдоль канала в направлении приложенной разности потенциалов, тогда как электроны преим. движутся (дрейфуют) по замкнутым траекториям вдоль азимута, в направлении, перпендикулярном E и H .

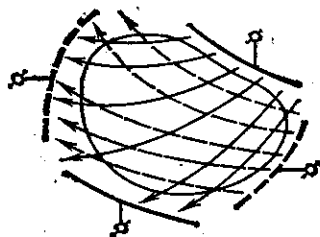


Рис. 2. Схематическое изображение траекторий ионов и электронов в плазменном объёме при «сильном» эффекте Холла; сплошные линии — ионы, штриховые — электроны.

Сущест. различие ионных и электронных траекторий приводит к тому, что сопряжение плазменных систем с электродами представляет собой весьма непростую проблему и часто требует сложных многоэлектродных систем, примером к-рых могут служить секциониров. электроды МГД-генераторов. Чтобы уменьшить возникающие здесь трудности, часто стремятся траектории той или иной группы частиц (обычно электронов) сделать замкнутыми.

Лит.: 1) Брагинский С. И., Явления переноса в плазме, в сб.: Вопросы теории плазмы, в. 1, М., 1983, с. 183; 2) Райзер Ю. П., Основы современной физики газоразрядных процессов, М., 1980.

ПРОВОДИМОСТЬ ЭЛЕКТРИЧЕСКАЯ — см. *Электропроводность*.

ПРОГРАММА (от греч. πρόγραμμα — объявление, распоряжение) — заданный набор действий и (или) правил, подлежащих выполнению (проверке) нек-рым исполнителем, обычно автоматич. устройством, чаще всего ЭВМ; предписание, алгоритм. П. выглядит как конечная совокупность команд (инструкций), каждая из к-рых предписывает исполнителю выполнить нек-рую элементарную операцию над данными, хранящимися в памяти исполнителя (см. *Памяти устройства*). Последовательность исполнения П. определяется тем, что любая текущая команда, кроме завершающей, указывает однозначно на команду П., к-рая должна выполняться после текущей. Команды ветвления (усл. переходы) осуществляют выбор одного из нескольких (указанных в команде) продолжений на основании проверки условий, определяющих свойства данных, упоминаемых в команде. Кроме того, возможно многократное выполнение отд. команд. Поэтому последовательность выполняемых команд и длина этой последовательности при исполнении П. могут варьироваться, однозначно определяясь входными данными. Для П., состоящей из набора действий, её алгоритм заранее определён, в отличие от П., состоящей из набора правил, когда её алгоритм определяется самим исполнителем в процессе выполнения П. Т. о., П. является конечным объектом, к-рый побуждает исполнителя закономерно реагировать на потенциально бесконечное разнообразие входных данных.

Лит.: Математический энциклопедический словарь, М., 1988, с. 494; Язык компьютера, пер. с англ. М., 1989.

ПРОГРАММИРОВАНИЕ — 1) процесс составления программы, плана действий. 2) Раздел информатики, изучающий методы и приёмы составления программ. С долей условности П. как дисциплина разделяется на: теоретическое, изучающее матем. абстракции программ (как объектов с определ. логич. и информац. структурой) и способы их построения; системное, имеющее дело с разработкой *программного обеспечения ЭВМ*, т. е. программных комплексов массового и длительного использования; прикладное, обслуживающее конкретные применения ЭВМ во всём их разнообразии.

Составление программы является творческой задачей, т. к. поиски способа достижения даже чётко сформулированной цели в общем случае требуют выработки или привлечения дополнит. знания. В нек-рых частных случаях возможно нахождение более систематической и формальной процедуры П. Так, если задание на П. уже сформулировано в виде алгоритма (точное описание последовательности действий, направленных на решение поставленной задачи), то П. сводится к переводу (трансляции) с языка записи алгоритма (см. *Языки программирования*) к языку, непосредственно воспринимаемому исполнителем (напр., ЭВМ). В нек-рых матем. моделях задача перевода решается исчерпывающе. П. включает поиски систематич. процедур перевода записей алгоритма в программы или создание программ по условиям задачи и дополнит. информации.

Методика П. уделяет особое внимание исходным спецификациям (полной и точной формулировке задачи, к-рую должна решать ЭВМ), поскольку умелое использование заложенной в спецификации информации позволяет придать П. более достоверный характер. Важным аспектом П. является забота о чёткой структуре программы, обеспечивающей проверку её правильности, а главное — выделение и изоляцию тех фрагментов программы, дальнейшая детализация к-рых требует привлечения дополнит. знаний. Ещё одним средством проверки правильности уже составленной программы является её отладка, т. е. систематич. испытание программы на ЭВМ и сравнение производимого эффекта с ожидаемым. Хотя на практике отладка является преимуществ. способом проверки программ, теоретически она не может быть исчерпывающей, т. к. установление правильности программ путём конечной системы испытаний может быть достигнуто только для узкого класса задач.

Различают следующие методы П. Синтезирующее П. — полное построение программы по заданной спецификации задачи или по общему алгоритму её решения. Структурное П. является комбинацией модульного, восходящего и нисходящего П. Модульное П. опирается на библиотеку модулей (программ с заданными описаниями входных и выходных данных) и состоит в выборе подходящих модулей и в их быстрой (иногда автоматизированной) сборке в результирующую программу. Нисходящее П. решает поставленную задачу путём её последоват. детализации с помощью отд. модулей, в о с о д я щ е — в обратном порядке путём укрупнения модулей (от более детализированных к менее). Конкретизирующее П. предполагает существование универсальной программы, решающей любую задачу данного класса, и состоит в адаптации универсальной программы к особенностям решаемой задачи. В результате получается либо более простая программа, либо используется меньшее кол-во ресурсов, чем в общем случае. На практике применяются комбинации всех видов П.

Лит.: Бауэр Ф. Л., Гроа Г., Информатика, пер. с нем., 2 изд., ч. 1—2, М., 1990; Любимский Э. З., Мартынюк В. В., Трифонов Н. П., Программирование, М., 1980; Мейер В., Бодуэн К., Методы программирования, пер. с франц., т. 1—2, М., 1982; Математический энциклопедический словарь, М., 1988, с. 493—96, 836.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ — организованная совокупность программ постоянного употребления, ориентирующая ЭВМ на тот или иной класс