

Министерство образования и науки Российской Федерации

Воронежский государственный университет

**Использование среды MAPLE
для решения задач квантовой механики**

Методические указания к специальному практикуму

«Решение задач по квантовой механике»

для студентов физического факультета дневного отделения,

обучающихся по направлению 010400 «Физика»

Составитель: В. В. Чернушкин

Воронеж 2005

Рекомендовано к опубликованию решением научно-методического совета
физического факультета от 17 февраля 2005 г., протокол №3.

Составитель: Чернушкин В.В.

Методические указания подготовлены на кафедре теоретической физики
физического факультета Воронежского государственного университета

Предлагаемые методические указания представляет собой руководство для решения задач по курсу квантовой механики с использованием персональных компьютеров. В указаниях приводятся примеры организации квантово-механических расчетов с использованием среды MAPLE. Цель работы – дать студентам первоначальные сведения о среде компьютерной математики MAPLE, научить их моделировать задачи квантовой механики различной сложности на компьютере, выполнять различные математические операции и аналитические преобразования.

Материал методических указаний можно условно разбить на две части. В первой части рассматриваются общие вопросы организации работы системы MAPLE и содержится краткий обзор основных возможностей системы и встроенных функций. Вторая часть пособия посвящена рассмотрению конкретных задач квантовой механики и некоторых смежных вопросов.

Указания адресованы студентам 4-го курса кафедры теоретической физики, а также могут быть полезны студентам, аспирантам, преподавателям и специалистам, использующим пакет MAPLE в практической деятельности.

Содержание

<i>MAPLE – универсальная система обеспечения научных расчетов</i> _____	4
<i>Обзор интерфейса и основные функции пакета Maple</i> _____	5
Основы работы с Maple _____	6
Стандартные математические функции системы MAPLE _____	6
Символьные вычисления _____	9
Аналитические преобразования _____	9
Решение уравнений _____	14
Графические возможности системы _____	16
Основы программирования в среде MAPLE _____	17
Условное ветвление _____	18
Циклы for и while _____	18
Функции пользователя _____	20
Процедуры и процедуры-функции _____	20
Средства MAPLE для работы с векторами и матрицами _____	21
Основные средства ядра MAPLE для обработки матриц _____	22
Обзор модулей linalg и LinearAlgebra. _____	24
<i>Расчет коэффициентов теории угловых моментов</i> _____	25
<i>Нахождение решения стационарного уравнения Шредингера для различных квантовых систем</i> _____	27
Стоячие волны _____	27
Полупроницаемая перегородка _____	29
Прямоугольная потенциальная яма _____	32
Виртуальные уровни _____	34
Дираковская потенциальная гребенка (периодический потенциал) _____	36
Гармонический осциллятор _____	38

Частица внутри потенциальной сферы _____	39
Сферически симметричная прямоугольная яма конечной глубины _____	41
Задачи для самостоятельного решения _____	42
<i>Диаманитные состояния атома водорода</i> _____	43
Разложение волновых функций по сферическому кулоновскому базису _____	43
Магнитоиндуцированные поправки к интенсивностям линий серии Лаймана _____	46
<i>Библиография</i> _____	50

MAPLE – универсальная система обеспечения научных расчетов

MAPLE – система компьютерной математики, рассчитанная на широкий круг пользователей. До недавнего времени ее называли системой компьютерной алгебры, что указывало на особую роль символьных вычислений и преобразований, которые способна осуществлять эта система. Но такое название сужает сферу применения системы, так как на самом деле способна выполнять быстро и эффективно не только символьные, но и численные расчеты, сочетая это с превосходными средствами графической визуализации и подготовки электронных документов.

По мере своего распространения Maple 7 становится полезной для многих пользователей ПК, вынужденных в силу обстоятельств (работа, учеба, хобби) заниматься математическими вычислениями и всем, что с ними связано. А все это простирается от решения учебных задач в вузах до моделирования сложных физических объектов, систем и устройств, и даже создания художественной графики (например, фракталов).

Maple — типичная интегрированная система. Она объединяет в себе:

- мощный язык программирования (он же язык для интерактивного общения с системой);
- редактор для подготовки и редактирования документов и программ;
- современный многооконный пользовательский интерфейс с возможностью работы в диалоговом режиме;
- мощную справочную систему со многими тысячами примеров;
- ядро алгоритмов и правил преобразования математических выражений;
- численный и символьный процессоры;
- систему диагностики;
- библиотеки встроенных и дополнительных функций;
- пакеты функций сторонних производителей и поддержку некоторых других языков программирования и программ.

Основой для работы с символьными преобразованиями в Maple является ядро системы. Оно содержит сотни базовых функций и алгоритмов символьных преобразований. В новых реализациях объем ядра достигает 6-7 Мбайт. Имеется

также основная библиотека операторов, команд и функций. Многие встроенные в нее функции, как и функции ядра, могут использоваться без какого-либо объявления, другие нуждаются в объявлении. Кроме того, имеется ряд подключаемых пакетов (`packages`).

Дополнительные функции из пакетов могут применяться после объявления подключения пакета с помощью команды `with(name)`, где `name` — имя применяемого пакета. Общее число функций, с учетом встроенных в ядро и размещенных в пакетах в системе Maple V R4 приближается к 2500, в реализации R5 — к 2700, а в Maple 6 и Maple 7 оно уже превышает 3000. Это означает, что большинство задач может решаться в режиме прямого диалога с системой без использования каких-либо средств программирования.

Maple имеет входной язык программирования сверхвысокого уровня, ориентированный на решение математических задач практически любой сложности. Он служит для задания системе вопросов или, говоря иначе, задания входных данных для последующей их обработки. Это язык интерпретирующего типа и по своей идеологии напоминает добрый старый Бейсик. И такое сходство вовсе не недостаток, а огромное достоинство — ведь именно с Бейсика начался подлинный диалог пользователя напрямую с компьютером! Входной язык имеет большое число заранее определенных математических и графических функций, а также обширную библиотеку, подключаемую по мере необходимости.

Кроме того, Maple содержит язык процедурного программирования — Maple-язык, включающий вполне традиционные средства структурирования программ: операторы циклов, операторы условных и безусловных переходов, операторы сравнения, логические операторы, команды управления внешними устройствами, функции пользователя, процедуры и т. д.

Maple-язык также включает в себя все команды и функции входного языка, ему доступны все специальные операторы и функции. Многие из них являются весьма серьезными программами, например символьное дифференцирование, интегрирование, разложение в ряд Тейлора, построение сложных трехмерных графиков и т. д.

Языком реализации системы Maple является язык Си, на котором написано ядро системы, содержащее тщательно оптимизированные процедуры. Большинство же функций, которые содержатся в пакетах, написаны на Maple-языке, благодаря чему их можно модифицировать и даже писать свои собственные библиотеки. По разным оценкам, лишь от 5 до 10 % средств Maple создано на языке реализации — все остальное написано на Maple-языке.

Синтаксис структурных операторов языка Maple напоминает смесь Бейсика и Паскаля. Это облегчает знакомство с ним тем, кто имеет хотя бы начальный опыт программирования на этих языках. По близким к Бейсику правилам (и при помощи общепринятых математических сокращений) выполняется и ввод математических выражений в диалоговом режиме работы с системой.

Обзор интерфейса и основные функции пакета Maple

Рассмотрим теперь основные элементы интерфейса системы MAPLE, основы ее справочной системы и некоторые функции, которые далее будем

использовать при моделировании решений конкретных физических и математических задач.

Как у всех Windows-приложений, интерфейс Maple имеет ряд характерных элементов, таких как строка заголовка (сверху); строка главного меню; главная панель инструментов; контекстная панель инструментов, вид которой зависит от режима работы с Maple; окно ввода и редактирования документов; строка состояния (в самом низу окна). Пользовательский интерфейс Maple позволяет готовить документы, содержащие одновременно команды входного языка (с возможным преобразованием их в естественную математическую форму), результаты вычислений в виде обычных математических формул и графические данные. Это обеспечивает понятное представление исходных данных и результатов вычислений, а также удобство их повторного использования.

Необходимо отметить и прекрасно реализованную справочную систему Maple. Преодолев первые трудности общения с системой, пользователь быстро осваивает систему справки, которая позволяет без какой-либо бумажной документации получить исчерпывающую информацию о любом операторе, функции или пакете (разумеется, на английском языке).

Основы работы с Maple

После загрузки и запуска системы можно начать диалог с ней, используя ее операторы и функции (с параметрами) для создания и вычисления математических выражений. Работа осуществляется в режиме интерпретатора, то есть в стиле: «задал вопрос, получил ответ». Вопросы и ответы занимают отдельные блоки, выделяемые в левой части квадратными скобками. Ввод выражений осуществляется по обычным правилам, общим для текстовых редакторов. Знак фиксации конца выражения ; (точка с запятой) указывает, что результат его вычисления должен быть выведен на экран, а знак : (двоеточие) отменяет вывод и может использоваться как знак разделителя при записи нескольких выражений в одной строке. Клавиши перемещения курсора позволяют передвигаться по ранее введенным строкам на экране. Для окончания ввода вопроса и получения ответа необходимо нажать клавишу Enter.

Стандартные математические функции системы MAPLE

Как во всех современных языках программирования высокого уровня, основным средством представления данных в среде MAPLE являются выражения и функции. Выражения строятся по общим правилам из имен функций, операций, констант и переменных. Сразу придется обратить особое внимание на то, что прописные и строчные буквы в идентификаторах считаются различимыми, то есть имена Temp, temp, TEMP обозначают три различных объекта. Тип переменных обычно определяется во время первого обращения и может быть легко изменен. Для записи вещественных чисел используется десятичная точка, в противном случае число рассматривается как целое. Для записи операций с числами используются традиционные обозначения: +, -, *, /, ^, **. Последние две операции обрабатывают возведение в степень. Приоритет операций также традиционный и может быть изменен при помощи круглых скобок, в том числе многоуровневых:

```
> 2+2*20;
(2+2)*20;
```

42

80

Результат целочисленных операций MAPLE по возможности представляет в виде точного значения или выражения. Чтобы получить результат в форме с плавающей точкой, необходимо ввести в исходное выражение параметр с плавающей точкой:

```
> 2^100;
```

1267650600228229401496703205376

```
> 33^33;
```

129110040087761027839616029934664535539337183380513

```
> 33.^33;
```

.1291100401 10⁵¹

В ядро системы MAPLE встроены многочисленные математические функции, основные из которых представлены в таблице 1.

Рассмотрим несколько примеров использования некоторых из представленных в таблице 1 функций:

```
> [factorial(10),10!];
```

[3628800, 3628800]

```
> iquo(234,5);
```

46

```
> irem(234,5);
```

4

```
> igcd(60,72);
```

12

```
> lcm(60,72);
```

360

```
> sin(Pi/4);
```

 $\frac{1}{2}\sqrt{2}$

```
> cosh(11);
```

cosh(11)

```
> cosh(11.0);
```

29937.07087

Существует возможность получения ссылки на последний полученный результат с помощью оператора %, а функция evalf(f) указывает, что считать

нужно численно в представлении с плавающей точкой. Так, для последнего примера можно записать:

```
> cosh(11);
evalf(%);
```

cosh(11)
29937.07087

и получить тот же результат.

Таблица 1. Основные математические функции системы Maple.

sin — синус;	arcsin — арксинус;
cos — косинус;	Arccos — арккосинус;
tan — тангенс;	arctan — арктангенс;
sec — секанс;	Arcsec — арксеканс;
csc — cosecant;	Arccsc — арккосеканс;
cot — котангенс.	arccot — арккотангенс.
factorial (n) — факториал (альтернатива — n!);	argument — аргумент комплексного числа; 1
iquo(a,b) — целочисленное деление a на b;	Lcm(a,b) — наименьшее общее кратное.
irem(a,b) — остаток от деления a на b;	Abs — абсолютное значение числа;
polar — полярное представление комплексного числа	igcd(a,b) — наибольший общий делитель;
floor — наибольшее целое, меньшее или равное аргументу;	ceil — наименьшее целое, большее или равное аргументу;
signum (x) — знак x (-1 при x < 0, 0 при x = 0 и +1 при x > 0).	trunc — целое, округленное в направлении нуля;
log10 — логарифм по основанию 10;	round — округленное значение числа;
sinh — гиперболический синус;	Arcsinh — гиперболический арксинус;
cosh — гиперболический косинус;	arccosh—гиперболический арккосинус;
tanh — гиперболический тангенс;	arctanh — гиперболический арктангенс;
sech — гиперболический секанс;	arcsech — гиперболический арксеканс;
csch — гиперболический cosecant;	coth — гиперболический котангенс
log — логарифм по заданному основанию	conjugate — комплексно-сопряженное число;
exp — экспонента;	Pi — число ПИ
sqrt — квадратный корень	frac — дробная часть числа;
arctan(y,x) = argument(x+I*y) ¹	ln — натуральный логарифм;
Re — действительная часть комплексного числа;	Im — мнимая часть комплексного числа;

¹ Она возвращает угол радиус-вектора в интервале от -Pi до Pi при координатах конца радиус-вектора x и y (см. пример ниже):

Как уже отмечалось, целочисленные вычисления MAPLE проводит точно. В то же время, для операций с плавающей точкой необходимо устанавливать точность. Делается это с помощью системной переменной `Digits`, которая по умолчанию устанавливается в 10 знаков (в текущем представлении). Значение переменной `Digits` можно изменить вручную. Кроме того, точность текущего результата функции `evalf` можно задать явно во втором аргументе:

```
> Pi;
evalf(%);
Digits:=15;
evalf(Pi);
Digits:=10;
evalf(Pi,40);
```

π

3.141592654

Digits := 15

3.14159265358979

Digits := 10

3.141592653589793238462643383279502884197

Отметим также, что это далеко не исчерпывающий список стандартных математических функций системы MAPLE, общее число которых порядка 3000. Для получения более подробных сведений о стандартных функциях можно обратиться к справочной системе.

Символьные вычисления

Рассмотрим основные операции символьной обработки математических выражений, среди которых центральное место занимают мощные средства аналитических преобразований, программы решения уравнений и систем уравнений, а также другие полезные операции символьной математики.

Аналитические преобразования

Среди основных средств осуществления аналитических преобразований можно выделить подстановки, упрощение, расширение (раскрытие скобок) и факторизацию (разложение на множители) как наиболее часто встречающиеся в повседневной научно-исследовательской работе.

Подстановки позволяют заменить часть выражения в заданном выражении на другое выражение. Например,

```
> restart;
> a:=x^2+x^4+b;
```

$a := x^2 + x^4 + b$

```
> subs(x=1,a);
subs(x=b^2,a);
```

```
subs(b=-x^2,a);
```

$$2 + b$$

$$b^4 + b^8 + b$$

$$x^4$$

Для упрощения выражений MAPLE содержит функцию `simplify` — одну из самых мощных в системах символьной математики. Она предназначена для упрощения математических выражений. Например,

```
> simplify(x^2/x^5+x);
```

$$\frac{1 + x^4}{x^3}$$

```
> simplify(sin(x)^2+cos(x)^2);
```

$$1$$

Функция `simplify` — многоцелевая. Она обеспечивает упрощение математических выражений, выполняя множество типовых действий. В функцию `simplify` можно включать дополнительные параметры. Более подробные сведения о функции `simplify` можно получить в справочной системе.

Функция `expand` «расширяет» выражение `expr` и записывается в виде:

```
expand(expr, expr1, expr2, ..., exprn)
```

где `expr` — расширяемое выражение, `expr1`, `expr2`, ..., `exprn` — необязательные подвыражения — опции. В большинстве случаев эта функция особенно полезна для раскрытия скобок в выражении:

```
> d:=(1+x)^4+(1-x)^5;
```

```
expand(%);
```

$$d := (1 + x)^4 + (1 - x)^5$$

$$2 - x + 16 x^2 - 6 x^3 + 6 x^4 - x^5$$

Факторизация выражений может производиться с помощью функции `factor`:

```
> d2:=x^3+y^3;
```

```
factor(d2);
```

$$d2 := x^3 + y^3$$

$$(y + x) (x^2 - x y + y^2)$$

Суммирование членов некоторой последовательности может быть осуществлено с помощью функции `sum`:

```
> sum('k^2', 'k'=0..4);
```

$$30$$

```
> sum('k^2', 'k'=0..n);
```

$$\frac{1}{3}(n+1)^3 - \frac{1}{2}(n+1)^2 + \frac{1}{6}n + \frac{1}{6}$$

> `sum('k^2', 'k');`

$$\frac{1}{3}k^3 - \frac{1}{2}k^2 + \frac{1}{6}k$$

> `sum('a[k]*x^k', 'k'=0..4);`

$$(x^2 + x^4 + b)_0 + (x^2 + x^4 + b)_1 x + (x^2 + x^4 + b)_2 x^2 + (x^2 + x^4 + b)_3 x^3 + (x^2 + x^4 + b)_4 x^4$$

> `sum('a[k]*x^k', 'k'=0..n);`

$$\sum_{k=0}^n a_k x^k$$

> `Sum('k/(k+1)', 'k'=0..n) = sum('k/(k+1)', 'k'=0..n);`

$$\sum_{k=0}^n \frac{k}{k+1} = n + 1 - \Psi(n+2) - \gamma$$

> `sum('1/k!', 'k'=0..infinity);`

e

> `sum('1/k^2', 'k'=1..infinity);`

$$\frac{1}{6}\pi^2$$

> `sum('k/(k+1)', 'k'=RootOf(x^3-2));`

2

> `sum('k^(3/2)', 'k'=1..infinity);`

∞

Аналогично, функция `product` используется для нахождения произведений членов какой-либо последовательности:

> `product(k^2, k=1..4);`

576

> `product(k^2, k=1..n);`

$\Gamma(n+1)^2$

> `product(k^2, k);`

$\Gamma(k)^2$

> `product(a[k], k=0..4);`

$$(x^2 + x^4 + b)_0 (x^2 + x^4 + b)_1 (x^2 + x^4 + b)_2 (x^2 + x^4 + b)_3 (x^2 + x^4 + b)_4$$

> `product(a[k], k=0..n);`

$$\prod_{k=0}^n (x^2 + x^4 + b)_k$$

> Product(n+k, k=0..m) = product(n+k, k=0..m);

$$\prod_{k=0}^m (n+k) = \frac{\Gamma(n+m+1)}{\Gamma(n)}$$

> product(k, k=RootOf(x^3-2));
2

Функция limit вычисляет предел выражения при заданном условии:

> limit(sin(x)/x, x=0);

1

> limit(exp(x), x=infinity);

∞

> limit(exp(x), x=-infinity);

0

> limit(1/x, x=0, real);

undefined

> limit(exp(x^2)*(1-erf(x)), x=infinity);

0

> Limit(sin(x), x=0)=limit(sin(x), x=0);

$$\lim_{x \rightarrow 0} \sin(x) = 0$$

Дифференцирование функций осуществляется при помощи функции diff:

> diff(sin(x), x);

cos(x)

> diff(sin(x), y);

0

> diff(sin(x), x\$3);

-cos(x)

> diff(x*sin(cos(x)), x);

$$\sin(\cos(x)) - x \cos(\cos(x)) \sin(x)$$

> diff(tan(x), x);

$1 + \tan(x)^2$

> Diff(tan(x), x);

$\frac{\partial}{\partial x} \tan(x)$

```
> Diff(tan(x), x) = diff(tan(x), x);
```

$$\frac{\partial}{\partial x} \tan(x) = 1 + \tan(x)^2$$

```
> diff(f(x), x);
```

$$\frac{\partial}{\partial x} f(x)$$

```
> diff(f(x,y), x,y);
```

$$\frac{\partial^2}{\partial y \partial x} f(x, y)$$

```
> diff(f(x,y), x,y) - diff(f(x,y), y,x);
```

$$0$$

```
> diff(g(x,y,z), x,z,z);
```

$$\frac{\partial^3}{\partial z^2 \partial x} g(x, y, z)$$

Интегрирование функций осуществляется при помощи функции int:

```
> int( sin(x), x );
```

$$-\cos(x)$$

```
> int( sin(x), x=0..Pi );
```

$$2$$

```
> int( x/(x^3-1), x );
```

$$\frac{1}{3} \ln(x-1) - \frac{1}{6} \ln(x^2+x+1) + \frac{1}{3} \sqrt{3} \arctan\left(\frac{1}{3}(2x+1)\sqrt{3}\right)$$

```
> int( exp(-x^2), x );
```

$$\frac{1}{2} \sqrt{\pi} \operatorname{erf}(x)$$

```
> int( exp(-x^2)*ln(x), x=0..infinity );
```

$$-\frac{1}{4} \sqrt{\pi} \gamma - \frac{1}{2} \sqrt{\pi} \ln(2)$$

```
> int( exp(-x^2)*ln(x), x );
```

$$\int e^{(-x^2)} \ln(x) dx$$

```
> series( %, x=0, 4 );
```

$$(\ln(x) - 1)x + \left(-\frac{1}{3} \ln(x) + \frac{1}{9}\right)x^3 + O(x^5)$$

```
> int( %, x );
```

$$\frac{1}{2} \ln(x) x^2 - \frac{3}{4} x^2 - \frac{1}{12} x^4 \ln(x) + \frac{7}{144} x^4 + O(x^6)$$

```
> int( exp(-x^2)*ln(x)^2, x=0..infinity );
```

$$\frac{1}{16} \pi^{(5/2)} + \frac{1}{8} \sqrt{\pi} \gamma^2 + \frac{1}{2} \sqrt{\pi} \gamma \ln(2) + \frac{1}{2} \sqrt{\pi} \ln(2)^2$$

Решение уравнений

Решение линейных и нелинейных уравнений и неравенств — еще одна важная область математического анализа. Maple имеет мощные средства для решения уравнений. Так, для решения линейных и нелинейных уравнений в аналитическом виде используется достаточно универсальная и гибкая функция `solve(eqn, var)` или `solve({eqn1, eqn2, ... }, {var1, var2, ... })`, где `eqn` — уравнение, содержащее функцию ряда переменных, `var` — переменная, по которой ищется решение. Если при записи `eqn` не используются знак равенства или знаки отношения, считается, что `solve` ищет корни уравнения `eqn=0`.

Функция `solve` старается дать решение в аналитическом виде. Это не означает, что ее нельзя использовать для получения корней уравнений в численном виде. Просто для этого придется использовать функции `evalf` или `convert`.

```
> solve( f=m*a, a );
```

$$\frac{f}{m}$$

```
> a:=x^2+x^4+b:
solve( {f=m*a}, {a} );
```

$$\{x^2 + x^4 + b = \frac{f}{m}\}$$

```
> f := proc(x) x-cos(x) end proc:
solve( f(x),x);
```

$$\text{RootOf}(_Z - \cos(_Z))$$

```
> eq := x^4-5*x^2+6*x=2;
```

$$eq := x^4 - 5x^2 + 6x = 2$$

```
> solve(eq,x);
```

$$1, 1, \sqrt{3} - 1, -1 - \sqrt{3}$$

```
> sols := [solve(eq,x)];
```

$$sols := [1, 1, \sqrt{3} - 1, -1 - \sqrt{3}]$$

```
> sols[1];
```

$$1$$

```
> evalf(sols);
```

```
[1., 1., .732050808, -2.732050808 ]
```

```
> sols := {solve(eq,x)};
           sols := { 1,  $\sqrt{3} - 1$ ,  $-1 - \sqrt{3}$  }

> sols[1];
           1

> subs( x=sols[1], eq );
           2 = 2

> solve(x^4+x+1,x);
           RootOf(_Z^4 + _Z + 1, index = 1), RootOf(_Z^4 + _Z + 1, index = 2),
           RootOf(_Z^4 + _Z + 1, index = 3), RootOf(_Z^4 + _Z + 1, index = 4)

> evalf({%});
           { .7271360845 - .9340992895 I, -.7271360845 + .4300142883 I,
           -.7271360845 - .4300142883 I, .7271360845 + .9340992895 I }
```

Если необходимо найти приближенное численное решение уравнения на заданном интервале, то можно воспользоваться функцией `fsolve`:

```
> fsolve( tan(sin(x))=1, x );
           .9033391108

> poly := 23*x^5 + 105*x^4 - 10*x^2 + 17*x:
fsolve( poly, x, -1..1 );
           -.6371813185, 0.

> fsolve( poly, x, maxsols=3 );
           -4.536168981, -.6371813185, 0.

> q := 3*x^4 - 16*x^3 - 3*x^2 + 13*x + 16:
fsolve(q, x, 1..2);
           1.324717957

> fsolve(q, x, 2..5);
fsolve(q, x, 4..8);
           5.333333333

> fsolve(q, x, complex);
           -.6623589786 - .5622795121 I, -.6623589786 + .5622795121 I,
           1.324717957, 5.333333333

> f := sin(x+y) - exp(x)*y = 0:
g := x^2 - y = 2:
fsolve({f,g},{x,y},{x=-1..1,y=-2..0});
           { x = -.6687012050, y = -1.552838698 }
```

Для решения обыкновенного дифференциального уравнения можно использовать функцию `dsolve`:

```
> ode1 := diff(y(x), x) - y(x)^2 + y(x) * sin(x) - cos(x);
```

$$ode1 := \left(\frac{\partial}{\partial x} y(x) \right) - y(x)^2 + y(x) \sin(x) - \cos(x)$$

```
> ans1 := dsolve(ode1);
```

$$ans1 := y(x) = \sin(x) - \frac{e^{(-\cos(x))}}{-CI + \int e^{(-\cos(x))} dx}$$

Для решения дифференциальных уравнений в частных производных требуется подключить внешний модуль **PDEtools**:

```
> restart;
```

```
> with(PDEtools);
```

```
PDE := x*diff(f(x,y), y) - diff(f(x,y), x) = f(x,y)^2*g(x)/h(y);
```

```
ans := pdsolve(PDE);
```

[*PDEplot, build, casesplit, charstrip, dchange, dcoeffs, declare, difforder, dpolyform, dsubs, mapde, separability, splitstrip, splitsys, undeclare*]

$$PDE := x \left(\frac{\partial}{\partial y} f(x, y) \right) - \left(\frac{\partial}{\partial x} f(x, y) \right) = \frac{f(x, y)^2 g(x)}{h(y)}$$

$$ans := f(x, y) = \frac{1}{\int \frac{g(-a)}{h\left(-\frac{1}{2}a^2 + y + \frac{1}{2}x^2\right)} da + {}_2F1\left(y + \frac{1}{2}x^2\right)}$$

Графические возможности системы

Система MAPLE обладает очень широкими возможностями для графического представления математической информации в виде двумерных и трехмерных графиков. Рассмотрим простейшие из этих средств. Чтобы построить двумерный график функции одной переменной можно, к примеру, записать:

```
> F := (x) -> x^2 * cos(x);
```

```
plot(F(x), x = -5..5);
```

$$F := x \rightarrow x^2 \cos(x)$$

Результат выполнения команды plot представлен на рисунке 1.

Чтобы построить трехмерный график функции двух переменных, можно воспользоваться функцией plot3d:

```
> F3 := (x, y) -> cos(x) * (x + y)^2;
```

```
plot3d(F3(x, y), x = -5..5, y = -5..5);
```

$$F3 := (x, y) \rightarrow \cos(x) (x + y)^2$$

Результат выполнения этой команды представлен на рисунке 2.

Основы программирования в среде MAPLE

Рассмотрим основные конструкции MAPLE-языка программирования. В частности, остановимся на способах задания функций пользователя, реализации управляющих конструкций (ветвления и циклы), объявлении процедур.

Рисунок 1

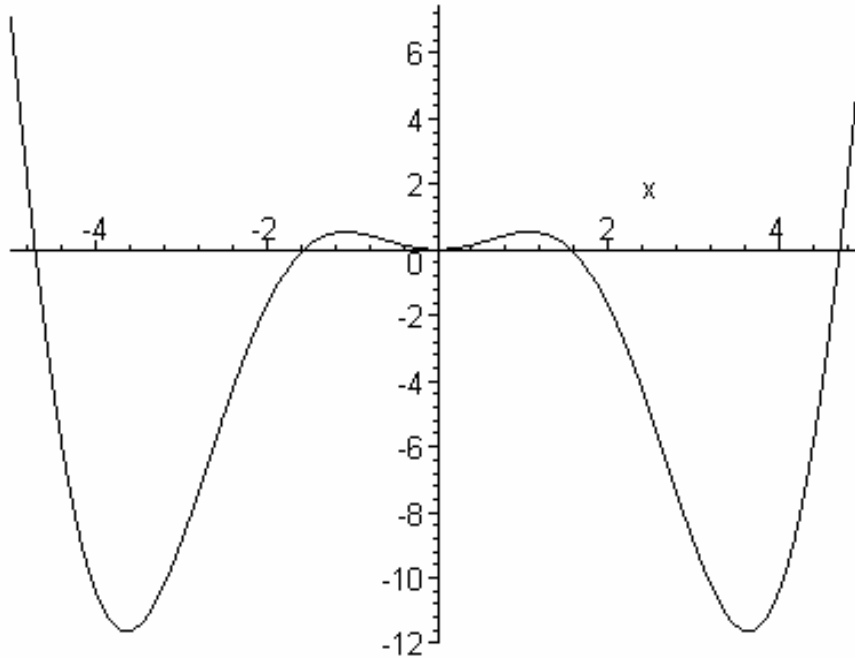
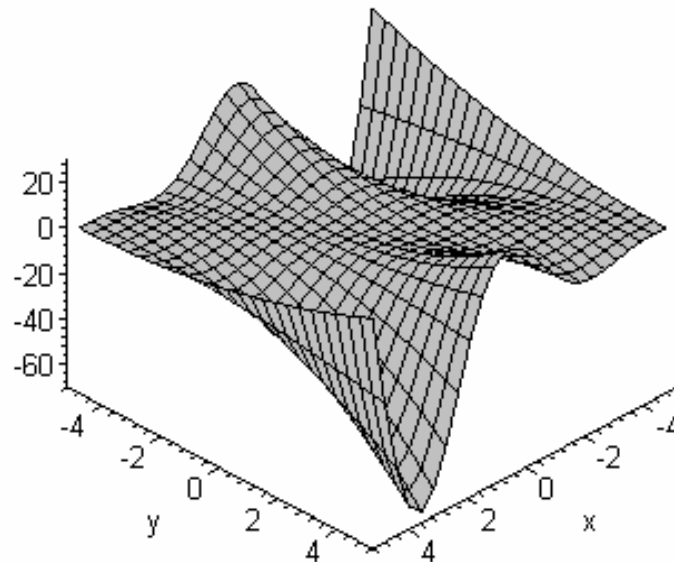


Рисунок 2



Для реализации структурности алгоритма на MAPLE-языке используются управляющие конструкции – ветвления и циклы.

Условное ветвление

Для подготовки разветвляющихся программ в Maple-язык программирования включен оператор `if`, позволяющий создавать следующую конструкцию:

```
if <Условие сравнения> then <Элементы>
| elif <Условие сравнения> then <Элементы> |
| else <Элементы> |
fi:
```

В вертикальных чертах `|` указаны необязательные элементы данной конструкции. Следующих два вида условных выражений чаще всего используются на практике:

```
if <Условие> then <Элементы 1> fi;
```

— если Условие выполняется, то исполняются Элементы 1, иначе ничего не выполняется;

```
if <Условие> then <Элементы 1> else <Элементы 2> fi;
```

— если Условие выполняется, то исполняются Элементы 1, иначе исполняются Элементы 2.

В задании условий используются любые логические конструкции со знаками сравнения (`<`, `<=`, `>`, `>=`, `=`, `<>`) и логические операторы `and`, `or` и `not`, конструкции с которыми возвращают логические значения `true` и `false`. Рассмотрим следующий простой пример:

```
> p:=-5;
if p<0 then print('Negative') fi;
      p := -5
      Negative
```

```
> p:=1;
if p<0 then print('Negative') fi;
      p := 1
```

Приведем еще один пример, показывающий более общую форму задания конструкции `if-then-else-fi`:

```
> p:=77;
if p<0 then print('Negative') else print('Positive') fi;
      p := 77
      Positive
```

К сожалению, функции на базе конструкции `if` не всегда корректно обрабатываются функциями символьной математики. Поэтому надо тщательно контролировать полученные в этом случае результаты.

Циклы `for` и `while`

В ряде случаев необходимо циклическое повторение выполнения выражения заданное число раз или до тех пор, пока выполняется определенное условие. Maple 7 имеет обобщенную конструкцию цикла, которая задается следующим образом:

```
| for <name>| |from <expr1>| |to <expr3>| |by <expr2>|
|while <expr4>| do Statement sequence> od;
```

Здесь name — имя управляющей переменной цикла, expr1, expr2 и expr3 — выражения, задающие начальное значение, конечное значение и шаг изменения переменной name, expr4 — выражение, задающее условие, пока цикл (набор объектов между словами do и od) будет выполняться.

В ходе выполнения цикла управляющая переменная меняется от значения expr1 до значения expr2 с шагом, заданным expr3. Если блок by <expr2> отсутствует, то управляющая переменная будет меняться с шагом +1 при expr1<expr2.

Следующий пример показывает задание цикла, у которого переменная цикла меняется от значения 1 до 10 с шагом 2:

```
> for i from 1 to 10 by 2 do print(i) od;
1
3
5
7
9
```

Шаг может быть и отрицательным, но если задать заведомо невыполнимое условие, например, expr1>expr2 и положительное значение шага, то цикл выполняться не будет. Цикл можно прервать с помощью дополнительного блока while <expr4>. Цикл с таким блоком выполняется до конца или до тех пор, пока условие expr4 истинно. Таким образом, конструкция цикла в Maple-языке программирования вобрала в себя основные конструкции циклов for и while. Есть еще одна, более специфическая конструкция цикла:

```
|for <name>| |in <expr1>| |while <expr2>| do statement
sequence> od:
```

Здесь expr1 задает список значений, которые будет принимать управляющая переменная name. Цикл будет выполняться, пока не будет исчерпан список и пока выполняется условие, заданное выражением expr2.

Также отметим, что возможна упрощенная частная конструкция цикла типа while:

```
while expr do statseq od:
```

Иногда бывает нужным пропустить определенное значение переменной цикла. Для этого используется оператор next (следующий). Другой оператор — break — прерывает выполнение фрагмента программы (или цикла). Действие этих операторов можно пояснить на примере:

```
> for i from 1 to 100 by 2 do
if i=5 then next fi;
print(i);
if i=7 then break fi;
od;
```

Функции пользователя

С помощью описанных управляющих конструкций можно писать программы любой сложности. Однако для реализации модульной структуры алгоритма требуется создание автономных модульных компонентов – функций и процедур.

Основной способ задания функций пользователя базируется на применении функционального оператора. При этом используется следующая конструкция:

name := (x, y, . . .) -> expr ;

После этого вызов функции осуществляется в виде name(x,y,...), где (x,y,...) — список формальных параметров функции пользователя с именем name. Переменные, указанные в списке формальных параметров, являются локальными. При подстановке на их место фактических параметров они сохраняют их значения только в теле функции.

> **restart ;**

> **m := (x, y) -> sqrt(x^2+y^2) ;**

$$m := (x, y) \rightarrow \sqrt{x^2 + y^2}$$

> **m(3, 4) ;**

5

> **m(1, 1) ;**

m(1, 1.0) ;

$$\sqrt{2}$$

1.414213562

Еще один способ задания функции пользователя базируется на применении функции unapply: name:=unapply(expr.var1.var2,...); Ниже приведен пример такого способа объявления функции пользователя:

> **m2 := unapply(sqrt(x^2+y^2), x, y) ;**

$$m2 := (x, y) \rightarrow \sqrt{x^2 + y^2}$$

> **m2(2, 1) ;**

$$\sqrt{5}$$

Процедуры и процедуры-функции

Наряду с функциями пользователя, важнейшими элементами структурного программирования в MAPLE являются процедуры и процедуры-функции. Процедурой называют модуль программы, имеющий самостоятельное значение и выполняющий одну или несколько операций, обычно достаточно сложных и отличных от операций, выполняемых встроенными операторами и функциями. Каждая процедура имеет свое уникальное имя и список параметров (он может

быть пустым). Процедуры вызываются так же, как встроенные функции, — указанием их имени со списком фактических параметров. При этом процедуры, в отличие от процедур-функций, обычно не возвращают каких-либо значений после своего исполнения, хотя могут присваивать значения входящим в них переменным. Процедуры-функции в ответ на обращение к ним возвращают некоторое значение. Они как бы являются новыми функциями, задаваемыми пользователем. Описанные ранее функции пользователя фактически являются процедурами-функциями с несколько упрощенной структурой. Простейшая форма задания процедуры следующая:

```
name :=proc(Параметры)
Тело процедуры
end;
```

Параметры процедуры задаются перечислением имен переменных, например `proc(x)` или `proc(x,y,z)`. С помощью знака `::` после имени переменной можно определить ее тип, например в объявлении `prog(n::Integer)` объявляется, что переменная `n` является целочисленной. При вызове процедуры выражением вида:

```
name(Фактические_параметры);
```

фактические параметры подставляются на место формальных. Несоответствие фактических параметров типу заданных переменных ведет к сообщению об ошибке и к отказу от выполнения процедуры.

Процедуры, которые возвращают значение результата в ответ на обращение к ним, во многом тождественны функциям. Будем называть их процедурами-функциями. Обычно процедура возвращает значение последнего выражения в ее теле или выражения, намеченного к возврату специальным оператором `RETURN::`.

В качестве примера рассмотрим процедуру вычисления модуля комплексного числа z — в данном случае это единственный параметр процедуры:

```
> modc:=proc(z)
evalf(sqrt(Re(z)^2+Im(z)^2)):
return %;
end;
modc := proc(z) evalf(sqrt(ℜ(z)^2 + ℑ(z)^2)); return % end proc
> modc(I+6);
6.082762530
```

Параметром оператора `RETURN` может быть любое выражение. В Maple не принято выделять процедуры-функции в какой-то отдельный класс. Действует правило — если не использован оператор `RETURN`, процедура возвращает значение последнего выражения в ее теле. Для устранения выдачи значений выражений внутри процедуры-функции после них просто надо установить знак двоеточия.

Средства MAPLE для работы с векторами и матрицами

Элементы векторов и матриц являются индексированными переменными, то есть место каждого элемента вектора определяется его индексом, а у матрицы

— двумя индексами. Обычно их обобщенно обозначают как i (номер строки матрицы или порядковый номер элемента вектора) и j (номер столбца матрицы). Допустимы операции вызова нужного элемента и присваивания ему нового значения:

- $V[i]$ — вызов i -го элемента вектора V ;
- $M[i, j]$ — вызов элемента матрицы M , расположенного на i -й строке в j -м столбце;
- $V[i]:=x$ — присваивание нового значения x i -му элементу вектора V ;
- $M[i,j]:=x$ — присваивание нового значения x элементу матрицы M .

Основные средства ядра MAPLE для обработки матриц

По структуре векторы и матрицы похожи на списки, но не полностью отождествляются с ними. В этом можно убедиться с помощью следующих примеров, в которых функция `type` используется для контроля типов множественных объектов (векторов и матриц):

```
> restart;
```

```
> M1 := [1, 2, 3, 4];
```

```
type(M1, vector);
```

$M1 := [1, 2, 3, 4]$

false

```
> V := convert(M1, vector);
```

```
type(V, vector);
```

$V := [1, 2, 3, 4]$

true

```
> M2 := [[1, 2], [3, 4]];
M := convert(M2, matrix);
```

```
type(M2, matrix);
```

```
type(M, matrix);
```

$M2 := [[1, 2], [3, 4]]$

$$M := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

false

true

Однако, используя функцию преобразования данных `convert`, можно преобразовывать одномерные списки в векторы, а двумерные — в матрицы. Функция `type` используется в следующих формах: `type(V, vector)` — тестирует аргумент V и возвращает `true`, если V — вектор, и `false` в ином случае; Аналогично, `type(M, matrix)` — тестирует аргумент M и возвращает `true`, если M — матрица, и `false` в ином случае. Здесь параметры `vector` и `matrix` используются для указания того, какой тип объекта проверяется.

Важное достоинство систем компьютерной алгебры, к которым относится и Maple 7, заключается в возможности выполнения аналитических (символьных) операций над векторами и матрицами.

Приведем примеры операций над векторами:

```
> evalm(V);
[1, 2, 3, 4]

> v[2]:=77;
V2:=77

> evalm(V);
evalm(3*V);
[1, 77, 3, 4]
[3, 231, 9, 12]
```

В этих примерах используется функция evalm(M), осуществляющая вычисление матрицы или вектора M.

Над матрицами с численными элементами можно выполнять разнообразные операции. Ниже приведены основные из них:

```
> evalm(M);
evalm(2+M);
evalm(M^2);
[ 1  2 ]
[ 3  4 ]
[ 3  2 ]
[ 3  6 ]
[ 7 10 ]
[15 22 ]

> M4:=evalm(M^(-1));
M4 := [ -2  1 ]
       [ 3  -1 ]
       [ 2  2 ]

> evalm(M4&*M);
[ 1  0 ]
[ 0  1 ]
```

Аналогичным образом можно осуществлять символьные операции над векторами и матрицами:

```
> s1:=array(1..2,1..2,[[a1,b1],[c1,d1]]);
S2:=evalm(S1^2);
S1 := [ a1  b1 ]
       [ c1  d1 ]
```

$$S2 := \begin{bmatrix} a1^2 + b1 c1 & a1 b1 + b1 d1 \\ c1 a1 + d1 c1 & b1 c1 + d1^2 \end{bmatrix}$$

Среди других функций для работы с матрицами полезно обратить внимание на функцию `map`, которая применяет заданную операцию (например, функции дифференцирования `diff` и интегрирования `int`) к каждому элементу матрицы. Пример такого рода дифференцирования матрицы приведен ниже:

```
> S3:=array(1..2,1..2,[[cos(x),sin(x)],[cos(x),-sin(x)]]);
S4:=evalm(S3^2);
map(diff,S4,x);
```

$$S3 := \begin{bmatrix} \cos(x) & \sin(x) \\ \cos(x) & -\sin(x) \end{bmatrix}$$

$$S4 := \begin{bmatrix} \cos(x)^2 + \sin(x) \cos(x) & \sin(x) \cos(x) - \sin(x)^2 \\ \cos(x)^2 - \sin(x) \cos(x) & \sin(x) \cos(x) + \sin(x)^2 \end{bmatrix}$$

$$\begin{bmatrix} -2 \sin(x) \cos(x) + \cos(x)^2 - \sin(x)^2 & -2 \sin(x) \cos(x) + \cos(x)^2 - \sin(x)^2 \\ -2 \sin(x) \cos(x) - \cos(x)^2 + \sin(x)^2 & \cos(x)^2 - \sin(x)^2 + 2 \sin(x) \cos(x) \end{bmatrix}$$

В результате возвращаются матрицы, каждый элемент которых представлен производной или интегралом. Аналогично можно выполнять над матрицами и другие достаточно сложные преобразования.

Обзор модулей `linalg` и `LinearAlgebra`.

В ядро Maple 7, как отмечалось, введены очень скромные и минимально необходимые средства для решения задач линейной алгебры. Основной упор в их реализации сделан на подключаемые пакеты. Основным из них, унаследованным от предшествующих реализаций системы, является пакет решения задач линейной алгебры `Linalg`. Это один из самых обширных и мощных пакетов в области решения задач линейной алгебры. Он содержит свыше ста функций, полный перечень которых можно получить при помощи следующей команды:

```
> with(linalg);
```

С деталями синтаксиса (достаточно разнообразного) для каждой из функций пакета можно ознакомиться в справочной системе Maple. Для этого достаточно использовать команду

```
> ? name;
```

где `name` — имя функции.

В последние годы разработчики систем символьной математики осознали, что малая скорость выполнения векторных и матричных операций при решении задач линейной алгебры оборачивается потерей заметной части рынка систем компьютерной математики. Поэтому в новых реализациях систем Maple и MATLAB была сделана ставка на использование давно апробированных быстрых алгоритмов линейной алгебры, предложенных создателями Number Algorithm Group (NAG). Эти алгоритмы издавна применяются на больших ЭВМ и суперкомпьютерах, обеспечивая ускорение численных матричных операций от

нескольких раз до нескольких десятков раз. В Maple 7 использование алгоритмов NAG является одной из первых отличительных черт новой версии системы. Оно реализуется новым пакетом `LinearAlgebra`. Для его загрузки используются следующие команды:

```
> restart;
with(LinearAlgebra);
```

Многие функции этого пакета повторяет по назначению функции более старого пакета `linalg`, но, в отличие от них, ориентированы на численные расчеты в формате обработки вещественных чисел, характерном для компьютерной платформы.

Более подробную информацию о работе в системе MAPLE и о ее использовании в различных областях физико-математических исследований можно получить в специальной литературе (см., например, [1]–[6]).

Расчет коэффициентов теории угловых моментов

При рассмотрении большинства задач квантовой механики, возникает необходимость расчета коэффициентов теории угловых моментов. Встроенных средств для осуществления таких расчетов MAPLE не имеет. Поэтому ниже мы рассмотрим один из вариантов создания таких функций. Обсуждаемые ниже функции находятся в пакете `AngleMoment`, составленном автором. Файл пакета доступен в Интернет по адресу:

www.tchernouchkine.narod.ru/maple/programs/anglemoment.m.

Вы можете скачать этот файл и подключить его к программам, испытывающим необходимость в поддержке функций теории углового момента. Для определенности будем считать, что файл `anglemoment.m` доступен локально и находится в корневом каталоге диска A: В этом случае для подключения пакета `AngleMoment` к программе, в ней необходимо выполнить следующую последовательность команд:

```
> read(`a:/anglemoment.m`);
with(AngleMoment);
```

После этого выводится список загруженных подпрограмм.

```
[Clb, Faza, W3jm, W6j, W9j, Way, kron, ph]
```

которые могут быть разделены на основные и вспомогательные. Описание этих функций содержится в таблицах 2 и 3.

После успешной загрузки внешнего модуля, подключенные функции можно использовать наряду со стандартными функциями системы MAPLE. Например,

```
> Clb(1,1,1,1,2,2);
```

1

```
> Clb(1,1,2,1,2,2);
```

$$\frac{1}{6}\sqrt{6}\sqrt{2}$$

```
> W6j(1,1,1,1/2,1/2,1/2);
```

> w9j(1,1,1,1,2,1,1,1,2);

Таблица 2 Основные функции пакета AngleMoment

Имя	Формат	Значение	Назначение
Clb	Clb(j1, m1, j2, m2, j3, m3)	$C_{j_1 m_1 j_2 m_2}^{j_3 m_3}$	Коэффициенты Клебша-Гордана
W3jm	W3jm(j1, m1, j2, m2, j3, m3)	$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$	3jm-символы Вигнера
W6j	W6j(j1, j2, j3, l1, l2, l3)	$\begin{Bmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l_3 \end{Bmatrix}$	6j-символы Вигнера
W9j	W9j(j11, j12, j13, j21, j22, j23, j31, j32, j33)	$\begin{Bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{Bmatrix}$	9j-символы Вигнера

Таблица 3 Вспомогательные функции пакета AngleMoment

Имя	Формат	Значение
ph	ph()	$(a)_k = \prod_{k=0}^{n-1} (a+k)$
Way	Way(j)	$\sqrt{2j+1}$
kron	kron(i, j)	$d_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$
Faza	Faza(n)	$(-1)^n$

Расчетные формулы для этих функций широко известны и довольно подробно описаны в специальной литературе (см., например, [8], [9], [10]). Нет необходимости приводить в данной работе исходные тексты функций модуля AngleMoment, так как в системе MAPLE пользователь может вызвать вывод листинга любой процедуры, написанной на MAPLE-языке. Предварительно требуется произвести соответствующую настройку интерфейса системы с помощью команды

> **interface(verboseproc=2, prettyprint=1, version);**

после чего просмотр листинга вызывается с помощью команды print:

```
print(Clb);
```

или

```
print(sin);
```

Исключение из этого правила составляет незначительное число процедур и функций системы MAPLE, написанных на других языках (как правило, С, Fortran) и включенных в систему MAPLE в скомпилированном виде. Очевидно, исходные тексты этих программ MAPLE не помнит.

Нахождение решения стационарного уравнения Шредингера для различных квантовых систем

В данном разделе рассматривается применение системы MAPLE для решения некоторых конкретных задач квантовой механики из сборника [7]. Не повторяя рассуждения, приведенные в сборнике, где подробно обсуждается ход решения каждой задачи, мы попытаемся визуализировать решение каждой задачи, создать информационные модели рассматриваемых задач, удобные для исследования. В качестве вводного упражнения найдем решение уравнения Шредингера для свободного одномерного движения частицы, то есть, когда потенциал $U(x) = 0$. Для этого введем и запустим следующую программу:

```
> # Free
restart;
U:=(x)->0;
a:=1;
schr:=diff(psi(x),x,x)+2*m/h^2*(E-U(x))*psi(x)=0;
dsolve(schr);
```

После исполнения программа выдает следующий ответ:

$$U := 0$$

$$a := 1$$

$$schr := \left(\frac{\partial^2}{\partial x^2} \psi(x) \right) + \frac{2 m E \psi(x)}{h^2} = 0$$

$$\psi(x) = -C1 \sin\left(\frac{\sqrt{2} \sqrt{m} \sqrt{E} x}{h}\right) + -C2 \cos\left(\frac{\sqrt{2} \sqrt{m} \sqrt{E} x}{h}\right)$$

хорошо известный из теории.

Стоячие волны

Частица заключена между двумя непроницаемыми стенками, расположенными в точках $x = -a$ и $x = +a$. Найти собственные состояния и обсудить их свойства (Ф18²).

Так как между стенками потенциал отсутствует, то внутри этой области справедливо общее решение уравнения Шредингера для свободной частицы:

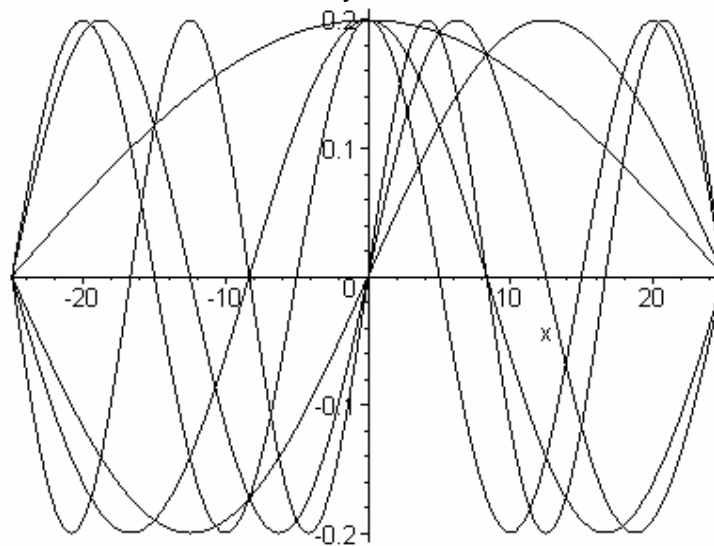
```
restart;
```

² Здесь и далее через Фхх обозначается номер задачи в задачнике [7]

```
> dsolve(diff(u(x), x, x) + k^2 * u(x) = 0);
```

$$u(x) = _C1 \sin(k x) + _C2 \cos(k x)$$

Рисунок 3



Задание граничных условий налагает ограничения на допустимые значения величины k . Исследуя выражения для нормированных волновых функций,

```
> u1 := (a, n, x) -> 1/sqrt(a) * sin(Pi * 2 * n / 2 / a * x);
```

```
u2 := (a, n, x) -> 1/sqrt(a) * cos(Pi * (2 * n - 1) / 2 / a * x);
```

$$u1 := (a, n, x) \rightarrow \frac{\sin\left(\frac{\pi n x}{a}\right)}{\sqrt{a}}$$

$$u2 := (a, n, x) \rightarrow \frac{\cos\left(\frac{1}{2} \frac{\pi (2n - 1) x}{a}\right)}{\sqrt{a}}$$

нетрудно проверить их нормировку

```
> int(u1(a, n, x)^2, x = -a..a);
```

```
int(u2(a, n, x)^2, x = -a..a);
```

1

1

Можно также получить явные выражения для этих функций,

```
> for n0 from 1 to 3 do u1(a, n0, x); u2(a, n0, x) od;
```

Для нахождения среднего значения импульса можно вычислить интегрированием

```
> int(u1(a, n, x) * diff(u1(a, n, x), x), x = -a..a);
```

```
int(u2(a, n, x) * diff(u2(a, n, x), x), x = -a..a);
```

0

0

Можно также получить графическое представление собственных функций.

Например, графики первых трех (четных и нечетных) собственных функций при $a = 25$ представлены на рисунке 3.

```
plot([u1(25,1,x),u1(25,2,x),u1(25,3,x),u2(25,1,x),u2(25,2,x),
),u2(25,3,x)],x=-25..25);
```

Полупроницаемая перегородка

Частица заключена между двумя непроницаемыми стенками, расположенными в точках $x = -a$ и $x = +a$. Кроме того, в точке $x = 0$ находится бесконечно узкая и бесконечно высокая полупрозрачная перегородка. Выяснить влияние перегородки на стационарные состояния (Ф19).

Как следует из приведенного в [7] решения, стационарные состояния рассматриваемой системы подразделяются на четные, удовлетворяющие уравнению (19.6а), и нечетные, определяемые уравнением (19.6б).

Для нечетных собственных функций из уравнения (19.6б) можно получить явное выражение $u_n^-(x) = \pm \frac{1}{\sqrt{a}} \sin\left(\frac{pnx}{a} + pn\right)$, $n = 1, 2, 3, \dots$

Построим графики первых трех собственных функций при $a = 1$ (см. рисунок 4):

```
> # Fluegge19;
```

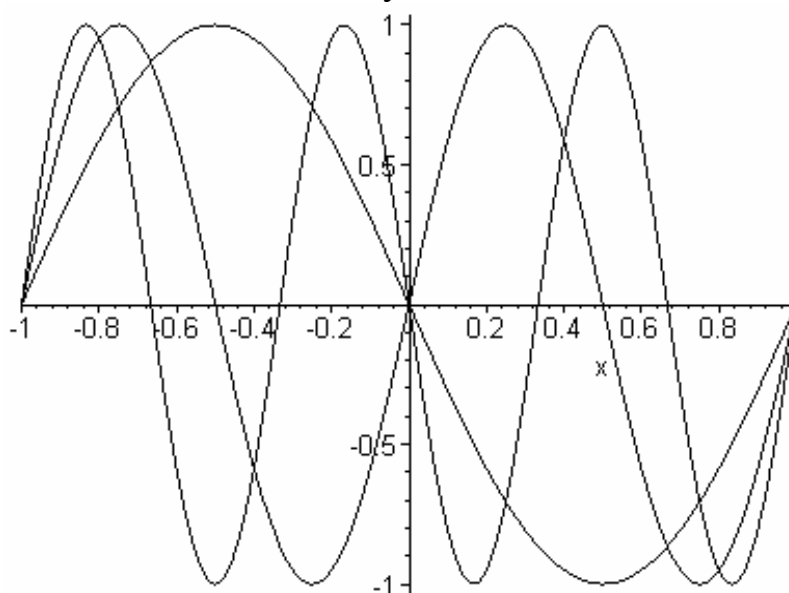
```
restart;
```

```
> u1:=(a,n,x)->1/sqrt(a)*sin(Pi*n*(x/a+1));
```

$$u1 := (a, n, x) \rightarrow \frac{\sin\left(\pi n \left(\frac{x}{a} + 1\right)\right)}{\sqrt{a}}$$

```
> plot([u1(1,1,x),u1(1,2,x),u1(1,3,x)],x=-1..1);
```

Рисунок 4



Все эти функции обращаются в нуль не только у стенок, но и на перегородке (для обеспечения непрерывности и нечетности).

Что касается четных функций, удовлетворяющих условию (19.6а), то для них собственные значения действительно можно получить только численным путем. Для этого нужно численно решить уравнение (19.6а), для заданного значения Ωa , причем каждое собственное значение k_n^+ лежит в интервале $p\left(n - \frac{1}{2}\right) < k_n^+ a < pn$. Зададим множество предполагаемых значений Ωa в переменной om:

```
> om:=[0,1/4,1/2,1,2,4,10,100,1000,1e+6];
```

```
om := [0, 1/4, 1/2, 1, 2, 4, 10, 100, 1000, .1 10^7]
```

и рассчитаем соответствующие им собственные значения величины $k_n^+ a$:

```
> n2:=5;
```

```
n3:=nops(om);
```

```
kna:=array(1..n3,1..n2);
```

```
oma:=array(1..n2);
```

```
for nz from 1 to n3 do
```

```
z:=op(nz,om):
```

```
for n from 1 to n2 do
```

```
kna[nz,n]:=fsolve(y*cot(y)=-z,y,Pi*(n-1)..Pi*n):;
```

```
od;
```

```
od;
```

Полученные значения сохранены в массиве kna и могут быть легко выведены:

```
> print(kna);
```

```
[1.570796327 , 4.712388980 , 7.853981634 , 10.99557429 , 14.13716694
1.715507153 , 4.764808915 , 7.885674079 , 11.01826000 , 14.15482692
1.836597203 , 4.815842318 , 7.917052685 , 11.04082982 , 14.17243207
2.028757838 , 4.913180439 , 7.978665712 , 11.08553841 , 14.20743673
2.288929728 , 5.086985094 , 8.096163603 , 11.17270587 , 14.27635292
2.570431560 , 5.354031841 , 8.302929183 , 11.33482558 , 14.40797112
2.862772588 , 5.760557933 , 8.708313831 , 11.70267808 , 14.73347234
3.110497702 , 6.221054828 , 9.331730126 , 12.44258102 , 15.55366297
3.138454210 , 6.276908481 , 9.415362876 , 12.55381746 , 15.69227228
3.141589512 , 6.283179024 , 9.424768536 , 12.56635805 , 15.70794756]
```

Полученные таким образом уровни энергии можно изобразить графически (см. рисунок 5) и проследить их изменение в зависимости от степени непрозрачности перегородки (от прозрачной к непрозрачной):

```
> plot([kna[a,1],kna[a,2],kna[a,3],kna[a,4],kna[a,5]],
a=1..n3);
```

Изобразим теперь графики собственных функций $u_n^+(a, b_n, x)$, $b_n = k_n^+ a$, $n = 1, 2, 3, \dots$ с положительной четностью

```
> u2:=(a,b,x)->-1/sqrt(a*(1-
sin(2*b)/(2*b)))*sin(b*(abs(x)/a-1));
```

$$u2 := (a, b, x) \rightarrow - \frac{\sin\left(b\left(\frac{|x|}{a} - 1\right)\right)}{\sqrt{a\left(1 - \frac{1}{2} \frac{\sin(2b)}{b}\right)}}$$

```
> for j1 to n2 do
plot([u2(1,kna[1,j1],x),u2(1,kna[2,j1],x),u2(1,kna[3,j1],x)
,u2(1,kna[4,j1],x),u2(1,kna[5,j1],x)
,u2(1,kna[6,j1],x),u2(1,kna[7,j1],x),u2(1,kna[8,j1],x),u2(1
,kna[9,j1],x),u2(1,kna[10,j1],x)
,x=-1..1);od;
```

Рисунок 5

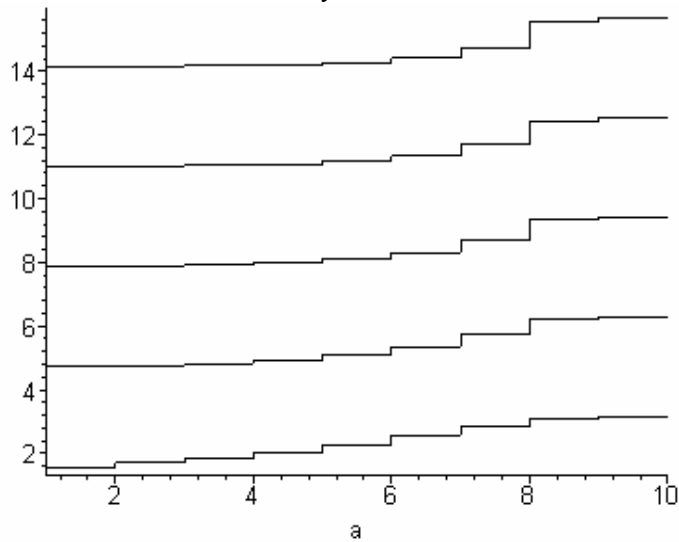
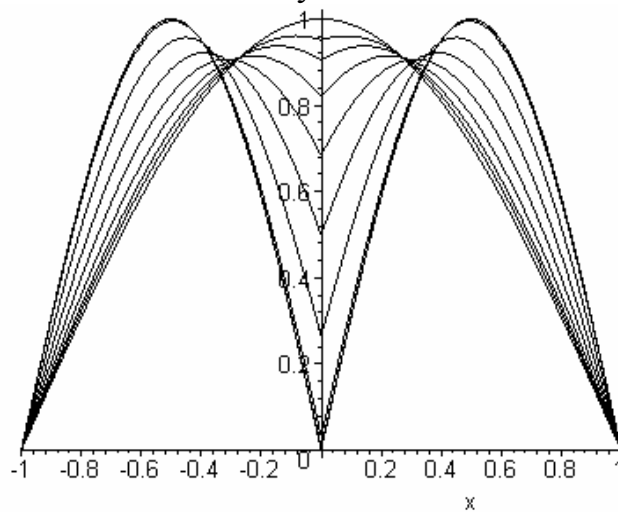


Рисунок 6



На рисунке 6 приведен график, соответствующий состояниям с наименьшим n для различных значений коэффициента непрозрачности (первый столбец вышеприведенной таблицы). В то же время, программа может нарисовать любую комбинацию графиков, в том числе на одном рисунке, что предоставляет

массу интересных возможностей при анализе теоретической информации, проведении вычислительных экспериментов, проверке решений на соответствие начальным условиям и во многих других ситуациях.

Прямоугольная потенциальная яма

Найти связанные состояния и соответствующие им собственные значения в случае прямоугольной потенциальной ямы (Ф25).

$$V(x) = \begin{cases} -U_0, & |x| < a \\ 0, & |x| > a \end{cases}$$

В отличие от предыдущей задачи, при рассмотрении потенциальной ямы конечной глубины учет граничных условий приводит к трансцендентным уравнениям (25.4а) и (25.4б) для нахождения собственных значений k_n^+ и k_n^- , которые могут быть решены численно:

```
> # Fluegge25;
```

```
restart;
```

```
> #+четные
```

```
f1 := (C, y) -> tan(y) - sqrt(C^2 - y^2) / y;
```

$$f1 := (C, y) \rightarrow \tan(y) - \frac{\sqrt{C^2 - y^2}}{y}$$

```
> #-нечетные
```

```
f2 := (C, y) -> cot(y) + sqrt(C^2 - y^2) / y;
```

$$f2 := (C, y) \rightarrow \cot(y) + \frac{\sqrt{C^2 - y^2}}{y}$$

Предварительно из общих свойств тангенса и котангенса можно определить число корней и интервалы, на которых эти корни находятся

```
> #for m=1, h=1
```

```
U0 := 100;
```

```
a := 2;
```

```
C := sqrt(2*U0)*a;
```

```
N1 := trunc(C/Pi+1);
```

```
N2 := trunc(C/Pi+1/2);
```

```
E_1 := matrix(2, N1);
```

```
E_2 := matrix(2, N2);
```

```
U0 := 100
```

```
a := 2
```

```
C := 20*sqrt(2)
```

```
N1 := 10
```

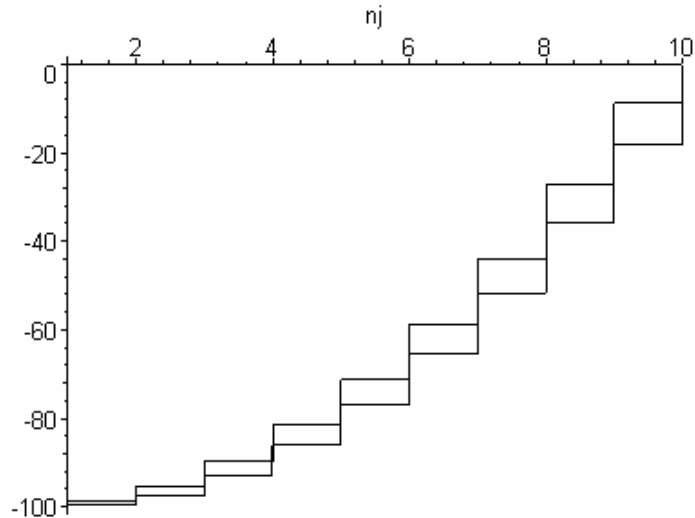
```
N2 := 9
```

```
E_1 := array(1 .. 2, 1 .. 10, [ ])
```


$E_2 := \text{array}(1 .. 2, 1 .. 9, [])$

Далее применяем функцию `fsolve` для поиска решения на каждом из интервалов.

Рисунок 7



Собственные значения записываем в специальный массив. Во вторую строку того же массива записываем значения соответствующих значений энергии. Расчет целесообразно вести отдельно для четных

```
> for n from 1 by 1 to N1 do
E_1[1,n]:=fsolve(f1(C,y1)=0, y1=Pi*(n-1)..min(Pi*(n-
1/2),C));
E_1[2,n]:=-U0*(1-(E_1[1,n]/C)^2);
od;
```

и нечетных состояний

```
> for n from 1 by 1 to N2 do
E_2[1,n]:=fsolve(f2(C,y1)=0, y1=Pi*(n-1/2)..min(Pi*n,C));
E_2[2,n]:=-U0*(1-(E_2[1,n]/C)^2);
od;
```

Полученные результаты можно распечатать

```
> print(E_1,E_2);
[1.517131861 , 4.550791796 , 7.582577338 , 10.61100940 ,
13.63418398 , 16.64937865 , 19.65219647 , 22.63424711 , 25.57408075
, 28.28316766 ]
[-99.71228886 , -97.41128675 , -92.81306512 , -85.92580994 ,
-76.76362840 , -65.34977382 , -51.72389674 , -35.96135720 ,
-18.24579920 , -.0078034000 ],
[3.034113726 , 6.067004400 , 9.097318243 , 12.12339822 ,
15.14299874 , 18.15271860 , 21.14663746 , 24.11201395 , 27.00539810
]
[-98.84926924 , -95.39893220 , -89.65485010 , -81.62790195 ,
-71.33619865 , -58.80985092 , -44.10246550 , -27.32634790 ,
-8.838559200 ]
```

или представить графически (смотрите рисунок 7)

```
> plot({E_1[2,nj],E_2[2,nj]},nj=1..N1);
```

Виртуальные уровни

Потенциальная полость между точками $x=0$ и $x=a$ ограничена справа полупрозрачной перегородкой так, что между волновыми функциями внутри и вне полости существует лишь слабая связь. Показать, что при большом коэффициенте непроницаемости существуют узкие полосы энергии, для которых упомянутая связь становится довольно сильной. В качестве примера разобрать случай $\Omega a/p = 50$. (Ф27).

Получив точные соотношения для фазового угла d (27.3) и амплитуды A (27.4) и (27.4a), можно непосредственно перейти к исследованию виртуальных уровней.

Введем обозначения $\Omega a = p$ и $ka = b$ и составим программу для расчета A и d .

Выражение для d получаем из (27.3) путем применения к этому соотношению серии символьных операций, включая символьное решение уравнения с помощью функции solve:

```
> # Fluegge27;
```

```
> restart;
```

```
s:=factor(expand(cot(b+d)-cot(b)));
```

$$s := -\frac{1 + \cot(b)^2}{\cot(b) + \cot(d)}$$

```
> s2:=subs(cot(d)=1/r,s);
```

$$s2 := -\frac{1 + \cot(b)^2}{\cot(b) + \frac{1}{r}}$$

Здесь введено обозначение $r = \tan(d)$ для искомой величины, явный вид которой можно получить как решение уравнения $s2 = 2p/b$ относительно r :

```
> dr:=solve(s2=2*p/b,r);
```

$$dr := -2 \frac{p}{b + b \cot(b)^2 + 2 p \cot(b)}$$

```
> dr2:=(b)->-2*p/(b+b*cot(b)^2+2*p*cot(b));
```

$$dr2 := b \rightarrow -2 \frac{p}{b + b \cot(b)^2 + 2 p \cot(b)}$$

Для нахождения экстремальных значений амплитуды, можно исследовать величину $\frac{1}{A^2} = A2(b, p)$ по $ka = b$.

```
A2:=(1+2*p/b*sin(2*b)+2*(p/b)^2*(1-cos(2*b)));
```

```
Ab:=factor(diff(A2,b)/2/p);
```

$$A2 := 1 + \frac{2 p \sin(2 b)}{b} + \frac{2 p^2 (1 - \cos(2 b))}{b^2}$$

$$Ab := \frac{-\sin(2 b) b + 2 \cos(2 b) b^2 - 2 p + 2 p \cos(2 b) + 2 p \sin(2 b) b}{b^3}$$

Очевидно, производная Ab есть периодическая по b функция, корни которой следует искать на интервале $(n-1)p \leq 2ka \leq np$. При заданном значении p можно произвести численный расчет собственных значений b_n , соответствующих максимальному или минимальному проникновению внешней волны внутрь полости:

Рисунок 8

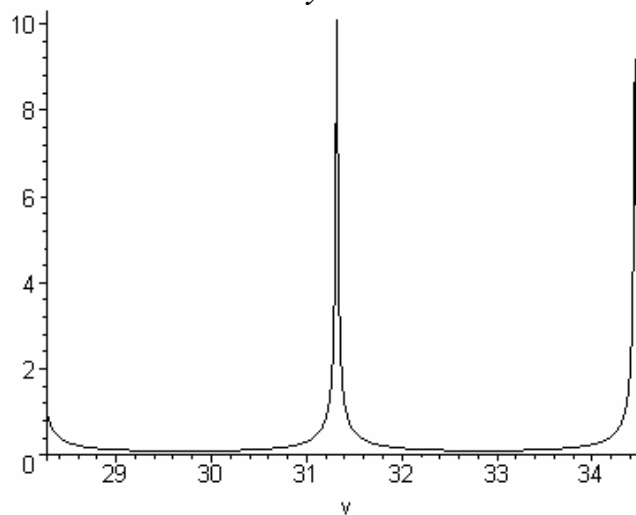
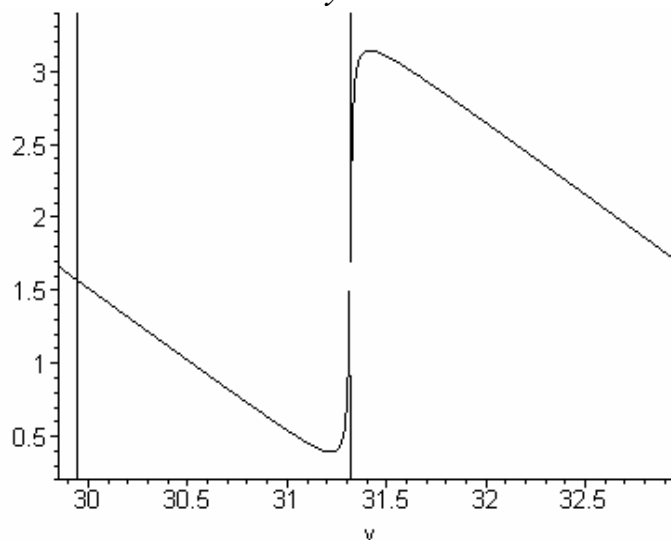


Рисунок 9



```
> p:=50*Pi;
bn:=proc(n)
fsolve(Ab=0,b,Pi/2*(n-1)..Pi/2*n);
end proc;
```

$$p := 50 \pi$$

`bn := proc(n) fsolve(Ab = 0, b, 1/2*π*(n - 1) .. 1/2*π*n) end proc`

`> bn(100);`

156.6874228

Аналогично вычисляются и абсолютные значения амплитуды A

`> A:=(b)->1/sqrt(1+2*p/b*sin(2*b)+2*(p/b)^2*(1-cos(2*b)));`

$$A := b \rightarrow \frac{1}{\sqrt{1 + \frac{2p \sin(2b)}{b} + \frac{2p^2(1 - \cos(2b))}{b^2}}}$$

Построим теперь графики полученных величин. На рисунке 8 изображен график функции $A(v)$:

`> plot(A(v), v=9*Pi..11*Pi);`

График величины $d = \arctan(dr2(v))$ получается без разрывов, если считать углы в диапазоне $0 \leq d \leq p$, в этом случае результат представлен на рисунке 9.

`plot({arctan(dr2(v)), Pi+arctan(dr2(v))}, v=19/2*Pi..21/2*Pi, 0.2..3.4);`

Дираковская потенциальная гребенка (периодический потенциал)

Дан периодический потенциал, образованный последовательностью d -функций Дирака. Определить зоны разрешенных значений энергии, если интервал между соседними особыми точками равен a (Ф29):

$$V(x) = \frac{\hbar^2}{m} \Omega \sum_{n=-\infty}^{\infty} d(x + na)$$

Наличие разрешенных энергетических зон определяется соотношением (29.7). Таким образом, определение границ энергетических зон в данном случае сводится к решению неравенства (29.8). Неравенство

$$\left(\cos y + \frac{w}{y} \sin y \right)^2 \leq 1$$

преобразуется к виду

$$\left(\frac{w^2}{y^2} - 1 \right) \sin^2 y + 2 \frac{w}{y} \sin y \cos y \leq 0$$

В свою очередь, уравнение

$$\left(\frac{w^2}{y^2} - 1 \right) \sin^2 y + 2 \frac{w}{y} \sin y \cos y = 0$$

имеет две группы решений. Во первых, его корнями являются точки $\sin y = 0$ или $y = n\pi$. Кроме того, указанному уравнению удовлетворяют точки, в которых

$\cot y = \frac{y^2 - w^2}{2yw}$. Такие точки имеются по одной на каждом из интервалов

$p(n-1) < y < pn$, каждый из этих корней может быть найден численно.

Квадраты соответствующих значений y определяют собственные значения энергии для границ зон.

```
> # Fluegge 29 with 28;
```

```
restart;
```

```
> P:=(w,y)->(cot(y)-y/2/w+w/2/y);
```

$$P := (w, y) \rightarrow \cot(y) - \frac{1}{2} \frac{y}{w} + \frac{1}{2} \frac{w}{y}$$

```
> w0:=4.0;
```

```
N:=10;
```

```
y_P:=matrix(4,N);
```

```
w0 := 4.0
```

```
N := 10
```

```
y_P := array(1 .. 4, 1 .. 10, [ ])
```

```
> for n from 1 to N do
```

```
y_P[1,n]:=fsolve(P(w0,y),y=(n-1)*Pi..n*Pi);
```

```
y_P[2,n]:=evalf(Pi*n);
```

```
y_P[3,n]:=y_P[1,n]^2;
```

```
y_P[4,n]:=y_P[2,n]^2;
```

```
od;
```

которые записываются в матрицу y_P и могут быть использованы в дальнейших расчетах.

```
> print(y_P);
```

```
[2.153747973 , 4.577859456 , 7.287194335 , 10.17397019 ,  
13.15666747 , 16.19232721 , 19.25912069 , 22.34541174 , 25.44459754  
, 28.55270584 ]
```

```
[3.141592654 , 6.283185308 , 9.424777962 , 12.56637062 ,  
15.70796327 , 18.84955592 , 21.99114858 , 25.13274123 , 28.27433389  
, 31.41592654 ]
```

```
[4.638630331 , 20.95679720 , 53.10320128 , 103.5096694 ,  
173.0978989 , 262.1914605 , 370.9137298 , 499.3174258 , 647.4275440  
, 815.2570108 ]
```

```
[9.869604404 , 39.47841761 , 88.82643963 , 157.9136706 ,  
246.7401101 , 355.3057584 , 483.6106159 , 631.6546817 , 799.4379569  
, 986.9604404 ]
```

Чтобы изобразить график, аналогичный Фиг.17 и Фиг 18 в [7] необходимо по известным k для разрешенных зон вычислить значения K из соотношения (29.7).

Это можно сделать, например, так

```
> KY0:=(w,y)->arccos(cos(y)+w/y*sin(y));
```

$$KY0 := (w, y) \rightarrow \arccos\left(\cos(y) + \frac{w \sin(y)}{y}\right)$$

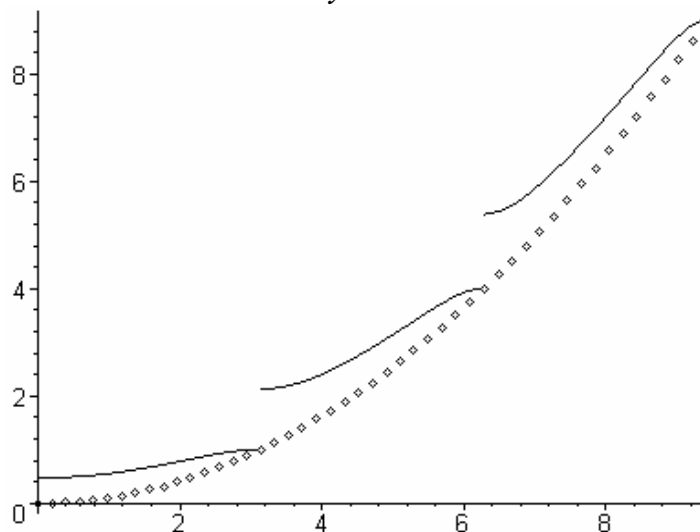
Графики на Фиг. 18 строятся следующим образом (как графики параметрических функций)

```
> plot([KY0(w0,y),y^2/Pi^2,
y=y_P[1,1]..y_P[2,1]]);
plot([Pi-KY0(w0,y)+Pi,y^2/Pi^2,
y=y_P[1,2]..y_P[2,2]]);
plot([KY0(w0,y)+2*Pi,y^2/Pi^2,
y=y_P[1,3]..y_P[2,3]]);
```

График, изображенный на Фиг. 17 получается сложением трех предыдущих и дополнительным нанесением параболы (другим стилем). Сложение графиков становится доступным после загрузки пакета plots и осуществляется функцией display (см. рисунок 10).

```
> with(plots):
G1:=plot([KY0(w0,y),y^2/Pi^2,y=y_P[1,1]..y_P[2,1]]):
G2:=plot([Pi-KY0(w0,y)+Pi,y^2/Pi^2,y=y_P[1,2]..y_P[2,2]]):
G3:=plot([KY0(w0,y)+2*Pi,y^2/Pi^2,y=y_P[1,3]..y_P[2,3]]):
G4:=plot(x^2/Pi^2,x=0..3*Pi,color=green,style=point):
display([G1,G2,G3,G4]);
```

Рисунок 10



Гармонический осциллятор

Найти собственные значения и собственные функции в случае осцилляторного потенциала (Ф30)

$$V(x) = \frac{1}{2}mw^2x^2.$$

Собственные функции, полученные в [7], выражаются через полиномы Эрмита, которые доступны в системе Maple после подключения пакета обработки ортогональных полиномов.

```
restart;
```

```
> with(orthopoly);
```

```
[G, H, L, P, T, U]
```

Теперь можно задать функции (30.19) в явном виде

```
> u := (n, lambda, x) -> (1/2^n/n! * sqrt(lambda/Pi))^(1/2) *
H(n, sqrt(lambda)*x) * exp(-1/2*lambda*x^2);
#30-19
```

$$u := (n, \lambda, x) \rightarrow \sqrt{\frac{\sqrt{\frac{\lambda}{\pi}}}{2^n n!}} H(n, \sqrt{\lambda} x) e^{(-1/2 \lambda x^2)}$$

Собственные значения энергии

```
> En := (n, omega) -> omega * (n+1/2);
#30-8
```

$$E_n := (n, \omega) \rightarrow \omega \left(n + \frac{1}{2} \right)$$

Проверка нормировки (при конкретном значении n) дает

```
> int(u(19,7,x)^2,x=-infinity..infinity);
#30-9
```

1

При желании можно вывести явный вид нескольких первых собственных функций

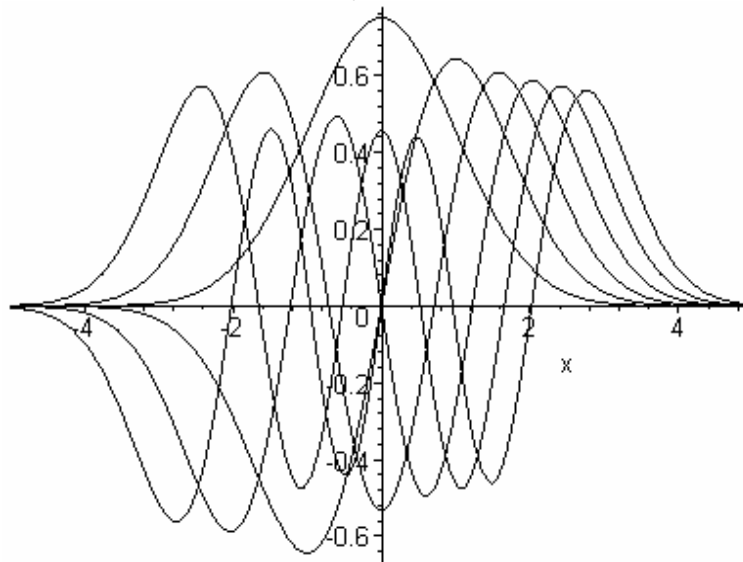
```
> for n1 from 0 to 10 do
```

```
print(n1, En(n1, lambda), factor(u(n1, lambda, x))) od;
```

и построить их графики, рисунок 11

```
plot([u(0,1,x), u(1,1,x), u(2,1,x), u(3,1,x), u(4,1,x), u(5,1,x)
], x=-5..5);
```

Рисунок 11



Частица внутри потенциальной сферы

Определить уровни энергии частицы, находящейся в сферически симметричном потенциальном ящике с бесконечными стенками и радиусом R (Ф62).

Для частицы внутри непроницаемой сферы собственные функции (62.9) выражаются через сферические функции Бесселя первого рода [11], [12].

```
> # Fluegge62, 63;
```

```
restart;
```

```
> Bes_j := (l, z) -> sqrt(Pi*z/2)*BesselJ(l+1/2, z);
```

```
Bes_n := (l, z) -> (-1)^(l+1)*sqrt(Pi*z/2)*BesselJ(-(l+1/2), z);
```

$$Bes_j := (l, z) \rightarrow \sqrt{\frac{1}{2} \pi z} \text{BesselJ}\left(l + \frac{1}{2}, z\right)$$

$$Bes_n := (l, z) \rightarrow (-1)^{(l+1)} \sqrt{\frac{1}{2} \pi z} \text{BesselJ}\left(-l - \frac{1}{2}, z\right)$$

которые могут быть выражены через тригонометрические функции:

```
> for n1 from 0 to 3 do print(simplify(Bes_j(n1, z))) od;
```

$$\frac{\sqrt{z} (-\cos(z) z + \sin(z))}{\sqrt{z^3}}$$

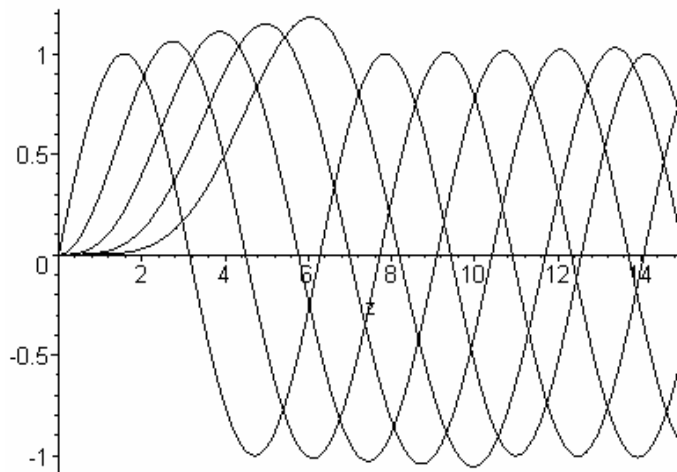
$$- \frac{\sqrt{z} (\sin(z) z^2 - 3 \sin(z) + 3 \cos(z) z)}{\sqrt{z^5}}$$

$$\frac{\sqrt{z} (\cos(z) z^3 - 15 \cos(z) z - 6 \sin(z) z^2 + 15 \sin(z))}{\sqrt{z^7}}$$

Построим графики нескольких простейших функций Bes_j (смотрите рисунок 12)

```
plot([Bes_j(0, z), Bes_j(1, z), Bes_j(2, z), Bes_j(3, z), Bes_j(4, z)
], z=0..15);
```

Рисунок 12



Для определения собственных значений необходимо найти нули функций Бесселя, которые можно записать в специальную матрицу для последующего использования, например, при рассмотрении следующей задачи.

```
> # Определить нули функции Бесселя
```

```
L_max:=16;
```



```

Nr_max:=16;
E_nr2:=array(0..L_max,1..Nr_max);
for nr1 from 1 to Nr_max do
E_nr2[0,nr1]:=Pi*nr1;
od;
for l1 from 1 to L_max do
for nr1 from 1 to Nr_max-l1 do
E_nr2[l1,nr1]:=fsolve(Bes_j(l1,z)=0,z=E_nr2[l1-1,nr1]..E_nr2[l1-1,nr1+1]);
od;
od;

```

Сферически симметричная прямоугольная яма конечной глубины

Для значений $l=0,1,2$ определить энергетические уровни связанных состояний в в сферически симметричной прямоугольной яме

$$V(r) = \begin{cases} -V_0, & r < R_0 \\ 0, & r > R_0 \end{cases}$$

Довести расчеты до численных результатов в том случае, когда характеристический параметр $2mV_0R^2/\hbar^2 = 100$ (Ф63).

В данной задаче помимо рассмотренных выше функций Бесселя первого рода фигурируют сферические функции Ханкеля мнимого аргумента, описывающие поведение частицы в классически недоступной области.

> # problem 63

> Han1_h := (l, z) -> sqrt(Pi*z/2)*HankelH1(l+1/2, z);

$$Han1_h := (l, z) \rightarrow \sqrt{\frac{1}{2}\pi z} \text{HankelH1}\left(l + \frac{1}{2}, z\right)$$

Можно запрограммировать характеристические функции (63.10)

> f_0 := (x, xi) -> -xi/sqrt(1-xi^2);

f_1 := (x, xi) -> x*xi/(1+xi^2/(1-xi^2)*(1+x*sqrt(1-xi^2)));

f_2 := (x, xi) -> x*xi*(1+x*sqrt(1-xi^2)+1/3*x^2*xi^2*(1-xi^2))/(1+x*sqrt(1-xi^2)*(1-1/3*x^2*xi^2*(1-xi^2)));

$$f_0 := (x, \xi) \rightarrow -\frac{\xi}{\sqrt{1-\xi^2}}$$

$$f_1 := (x, \xi) \rightarrow \frac{x\xi}{1 + \frac{\xi^2(1+x\sqrt{1-\xi^2})}{1-\xi^2}}$$

$$f_2 := (x, \xi) \rightarrow \frac{x\xi\left(1+x\sqrt{1-\xi^2} + \frac{1}{3}x^2\xi^2(1-\xi^2)\right)}{1+x\sqrt{1-\xi^2}\left(1-\frac{1}{3}x^2\xi^2(1-\xi^2)\right)}$$

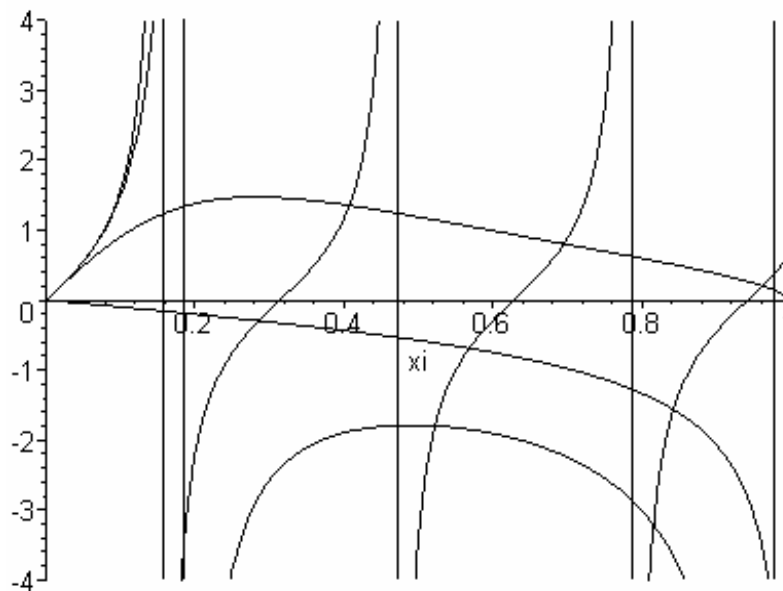
чтобы построить график, изображенный на фиг. 36 в [7].

```
plot([f_0(10,xi),f_1(10,xi),f_2(10,xi),tan(10*xi)],xi=0..1,
-4..4);
```

и численно определить собственные значения для связанных состояний, соответствующих точкам пересечения графиков на рисунке 13.

```
10*fsolve(f_0(10,xi)=tan(10*xi),xi=1*Pi/20..3*Pi/20);
10*fsolve(f_0(10,xi)=tan(10*xi),xi=3*Pi/20..5*Pi/20);
10*fsolve(f_0(10,xi)=tan(10*xi),xi=5*Pi/20..7*Pi/20);
10*fsolve(f_1(10,xi)=tan(10*xi),xi=1*Pi/20..3*Pi/20);
10*fsolve(f_1(10,xi)=tan(10*xi),xi=3*Pi/20..5*Pi/20);
10*fsolve(f_1(10,xi)=tan(10*xi),xi=5*Pi/20..7*Pi/20);
10*fsolve(f_2(10,xi)=tan(10*xi),xi=3*Pi/20..5*Pi/20);
10*fsolve(f_2(10,xi)=tan(10*xi),xi=5*Pi/20..7*Pi/20);
```

Рисунок 13



Задачи для самостоятельного решения

1(Ф36). Частица свободно движется по оси x между двумя абсолютно жесткими стенками, расположенными в точках $x = \pm a$. Аппроксимировать волновые функции основного и первого возбужденного состояний простейшими нормированными полиномами, имеющими те же нули, что и точные волновые функции, и сравнить точные значения с приближенными.

2(Ф37). Определить коэффициент отражения в поле потенциальной ступеньки $V(x) = \frac{1}{2}V_0 \left(1 + th \frac{x}{2a}\right)$

3(Ф38). Решить уравнение Шредингера с потенциалом

$$V(x) = \frac{1}{2}V_0 \left[\frac{k(k-1)}{\sin^2 ax} + \frac{l(l-1)}{\cos^2 ax} \right], \quad V_0 = \frac{\hbar^2 a^2}{m}$$

где $k > 1$ и $l > 1$. Ограничиться рассмотрением интервала $0 \leq x \leq p/2a$, на границах которого потенциал $V(x)$ обращается в бесконечность.

4(Ф40). Частица движется в однородном гравитационном поле над поверхностью земли, которая предполагается абсолютно упругой (например, стальной шарик, танцующий на стеклянной горизонтальной пластине). Проанализировать эту классическую задачу с точки зрения квантовой теории.

5(Ф64). Сферически симметричная потенциальная яма описывается потенциалом Вуда-Саксона

$$V(r) = -\frac{V_0}{1 + e^{\frac{r-R}{a}}}$$

где $a \ll R$. Определить энергию связанных состояний с $l = 0$.

6(Ф67). Решить уравнение Шредингера для связанных состояний электрона в поле бесконечно тяжелого точечного ядра с зарядом Ze .

7(Ф68). Решить уравнение Шредингера и определить уровни энергии в случае так называемого потенциала Хюльтена $V(r) = -V_0 \frac{e^{-r/a}}{1 - e^{-r/a}}$ при $l = 0$.

Диаманитные состояния атома водорода

Расчеты в рамках теории возмущений являются одним из основных методов, применяемых в современной теоретической физике. В частности, теория возмущений применима для широкого класса задач взаимодействия атомных объектов с постоянными внешними полями. В качестве примера рассмотрим построение волновых функций диаманитных состояний водородоподобного атома, определение диаманитных восприимчивостей и магнитоиндуцированных поправок к интенсивности линий с помощью теории возмущений высших порядков.

Разложение волновых функций по сферическому кулоновскому базису

Рассмотрим задачу нахождения магнитоиндуцированных поправок к вероятностям переходов из высоковозбужденных (вырожденных) состояний водородоподобных атомов в основное состояние [13].

Диаманитное взаимодействие

$$V_D = \frac{B^2}{12} r^2 \left(1 - C_{20} \left(\frac{\mathbf{r}}{r} \right) \right) \quad (1)$$

не является сферически симметричным. Под действием магнитного поля происходит перемешивание состояний с различными моментами одинаковой четности (в силу четности диаманитного оператора), и атом переходит в состояние с четностью $P = (-1)^m$, являющееся суперпозицией состояний с $l = m, m + 2, \dots, n - 1$ или в состояние с четностью $P = (-1)^{m+1}$ представляющее собой суперпозицию состояний с $l = m + 1, m + 3, \dots, n - 1$.

Высоковозбужденные уровни, в частности, $|nP\rangle$, представляют собой лишь компоненту земановского состояния $|nmI - \rangle$ с $m = 0, 1$ и отрицательной

четностью, в состав которого входят также состояния с моментами $l = 3, 5, \dots$. Эти уровни имеют различные квадратичные сдвиги, то есть диамагнитное взаимодействие полностью снимает вырождение в спектре водорода. Таким образом, перераспределение интенсивности дипольного излучения по диамагнитным состояниям $|nml\rangle$ в серии Лаймана (соответствующей переходам $|nP\rangle \rightarrow |1S\rangle$) определяется вкладом $|nP\rangle$ -волновой функции сферического кулоновского базиса. В нулевом приближении этот вклад не зависит от силы поля B , поскольку энергии вырожденных состояний совпадают и переход к диамагнитным состояниям связан лишь с изменением симметрии волновой функции.

С ростом B происходит как изменение матричного элемента $|nP\rangle$ -перехода, так и изменение вклада $|nP\rangle$ -состояния в диамагнитную волновую функцию. Кроме того, диамагнитное взаимодействие индуцирует переходы и из состояний с моментами $l = 3, 5, \dots$, также входящих в волновую функцию нулевого приближения. Для расчета этих поправок необходимо использовать теорию возмущений высших порядков для вырожденных состояний [14].

По аналогии с разложением диамагнитной энергии:

$$\Delta E_l = E_l - E_n = -\sum_{s=1}^{\infty} \frac{c_l^{(s)}}{2s!} B^{2s} \quad (2)$$

где $c_l^{(s)}$ - восприимчивость s -го порядка, запишем ряд для оператора диамагнитного взаимодействия $W_{ll'}$ в виде:

$$W_{ll'} = -\sum_{s=1}^{\infty} \frac{v_{ll'}^{(s)}}{2s!} B^{2s} \quad (3)$$

Матричный элемент радиационного перехода представим в виде

$$d_{if}(B) = d_{10} \left(1 + \sum_{s=1}^{\infty} q_{10}^{(s)} B^{2s} \right) \quad (4)$$

Коэффициенты a_l тоже следует разложить в степенные ряды, которые в отличие от величин (2), (3), обращающихся в нуль при $B \rightarrow 0$, начинаются с нулевой степени:

$$a_l = \sum_{s=0}^{\infty} a_l^{(s)} B^{2s} \quad (5)$$

и при $B = 0$ эти величины остаются конечными, $a_l \Big|_{B \rightarrow 0} = a_l^{(0)}$. В низшем порядке имеем:

$$\sum_{l'=m+p}^{l_{\max}} a_l^{(0)} \left(v_{ll'}^{(1)} - c^{(1)} d_{ll'} \right) = 0 \quad (6)$$

Таким образом, набор коэффициентов $a_l^{(0)}$ является собственным вектором, а $c^{(1)}$ - собственным значением матрицы диамагнитного взаимодействия $v_{ll'}^{(1)}$. Собственные значения этой матрицы представляют собой диамагнитные

восприимчивости первого порядка $c_l^{(1)}$, где $l=1, 2, \dots, K$, а K определяет кратность вырождения.

Рассмотрим MAPLE-программу, реализующую описанный алгоритм. Для использования матричных операций необходимо загрузить пакет `linalg`. Определяем начальные условия (главное и магнитное квантовые числа верхнего уровня) и несколько вспомогательных функций. Кратность вырождения k , определяющая размерность матриц определяется значением главного квантового числа верхнего уровня:

```
> restart;
> n:=50;
m:=0;
> with(linalg):
> kron := (i,j) -> if i=j then 1 else 0 fi;
> p4 := proc (a) options operator, arrow;
a*(a+1)*(a+2)*(a+3) end proc;
> k:=trunc(n/2);
```

Прежде всего, необходимо найти волновые функции нулевого приближения, точнее, коэффициенты их разложения по базису функций с определенным значением орбитального момента соответствующей четности (в отсутствии магнитного поля). Функция `V_1` определяет диамагнитную матрицу первого порядка $EV_1[(l+1)/2, (l'+1)/2] = v_{ll'}^{(1)}$.

```
> #The first-order on energies and zero-order on
coefficients
> V_1:=(n,l,le,m)->
> if le<1 then V_1(n,le,l,m)
> else if le=1 then ((l^2+l-1+m^2)/(2*l-
1)/(2*l+3))*n^2*(5*n*n+1-3*l*(l+1));
> else if le=1+2 then -5*n^2/2/(2*l+3)*(((l+1)^2-
m^2)*((l+2)^2-m^2)/(2*l+1)/(2*l+5)*(n^2-(l+1)^2)*(n^2-
(l+2)^2))^(1/2);
> else 0 fi fi fi;
> EV_1 := Matrix(k,k):
> #for i from 1 to k do for j from 1 to k do EV_1[i, j]
:=V_1(n,2*i-1,2*j-1,m) od od ;
> for i from 1 to k do for j from 1 to k do EV_1[i, j]
:=evalf(V_1(n,2*i-1,2*j-1,m),12) od od ;
> print(EV_1);
> det(EV_1);
> eigenvals(EV_1,'radical');
> y:=[eigenvects(EV_1)];
```

Собственные значения этой матрицы определяют диамагнитные восприимчивости первого порядка, а собственные векторы матрицы `EV_1` задают коэффициенты нулевого приближения, собранные в матрице `A_0`:

```

> A_0:=matrix(k,k,[]);
E_cor1:=vector(k,[]);
for la from 1 to k do
> E_cor1[la]:=y[la][1]:
> for lb from 1 to k do
> A_0[la,lb]:=y[la][3][][lb];
> od;
print(A_0[la,1])
> od;
print(A_0);

```

Таким образом, каждый элемент $A_0[la, (l+1)/2]=a_l^{(0)}(\lambda)$, соответствующий некоторому собственному значению $E_cor1[la]=c_l^{(1)}$ (диамагнитная восприимчивость первого порядка), определяет коэффициент разложения волновой функции диамагнитного состояния с номером l при невозмущенной функции с орбитальным моментом l , или вклад чистого состояния с моментом l в диамагнитное состояние с номером l . Проверить условие полноты можно, например, найдя сумму квадратов коэффициентов при определенном значении орбитального момента, например, $l=1$ для всех l .

```

> sum((A_0[p1,1])^2,p1=1..k);

```

Очевидно, такая сумма должна быть равной 1.

Магнитоиндуцированные поправки к интенсивностям линий серии Лаймана

Полученные значения $c_l^{(1)}$ и $a_l^{(0)}(l)$ используются для решения системы уравнений в следующем порядке по B^2 , которую можно представить в виде:

$$\sum_{l'=m+p}^{l_{\max}} a_{l'}^{(1)}(l) (v_{ll'}^{(1)} - c_l^{(1)} d_{ll'}) + \sum_{l'=m+p}^{l_{\max}} a_{l'}^{(0)}(l) (v_{ll'}^{(2)} - c_l^{(2)} d_{ll'}) = 0. \quad (7)$$

Домножая обе части этого уравнения на $a_l^{(0)*}(l)$ и суммируя по l , получим с учетом (6) и условия полноты, $\sum |a_l^{(0)}(l)|^2 = 1$, выражение для восприимчивости второго порядка:

$$c_l^{(2)} = \sum_{l'=m+p}^{l_{\max}} \sum_{l''=m+p}^{l_{\max}} a_{l'}^{(0)*}(l) a_{l''}^{(0)}(l) v_{ll''}^{(2)} \quad (8)$$

Таким образом, квадратичная диамагнитная восприимчивость определяется собственным вектором $a_l^{(0)}(l)$ матрицы первого порядка EV_1 и матричными элементами второго порядка EV_2, которые вычисляются при помощи функции V_2.

```

> #The Second order on energies and first-order on
coiefficients
> V_2:=(n,l,le,m)->
> if l>le then V_2(n,le,l,m)

```

```

> else if le=1 then B0:=n^6/240*(5*n^2*(97*n^2-
33*1*(1+1)+365)-12*(21*1^4+42*1^3+179*1^2+158*1+60));
B2:=n^6/(336*(2*1+3))*(n^2*(802*n^2-
1005*1*(1+1)+2000)+3*(175*1^4+350*1^3+523*1^2+348*1-480));
B4:=n^6/(1120*(2*1+3)*(2*1+5))*(15*n^2*(41*n^2+22*1*(1+1)-
185)-1001*1^3*(1+2)+3421*1^2+4422*1-2160);
B0+(3*m^2-1*(1+1))/((2*1-1))*B2+(3*(1^2+2*1-5*m^2)*(1^2-
5*m^2-1)-10*m^2*(4*m^2-1))/((2*1-1)*2*(2*1-3))*B4
> else if le=1+2 then
G0:=n^2*(429*1*(1+3)-802)+237*1^3*(1+6)+3145*1^2+3036*1-
2484;
G2:=5*(41*n^2+37*1^2+111*1-75);
-sqrt(((1+1)^2-m^2)*((1+2)^2-m^2)*(n^2-(1+1)^2)*(n^2-
(1+2)^2))/((2*1-
1)^2*(2*1+1)*(2*1+3)^2*(2*1+5)*(2*1+7)^2))*n^6/32*(G0+m^2*G
2);
else if le=1+4 then sqrt((p4(1+1-
m)*p4(1+1+m)*p4(n+1+1)*p4(n-1-
4))/((2*1+1)*(2*1+3)^2*(2*1+5)^2*(2*1+7)^2*(2*1+9)))*205*n^
6/128
> else 0 fi fi fi fi ;
>
> EV_2 := matrix(k,k);
for i from 1 to k do for j from 1 to k do EV_2[i, j]
:=evalf(V_2(n,2*i-1,2*j-1,m),12) od od;
print(EV_2);

```

Рассчитаем диамагнитные восприимчивости второго порядка $C_I^{(2)}$.

```

E_cor2:=vector(k,[]);
> for la from 1 to k do
> s:=0;
> for i from 1 to k do
> for j from 1 to k do
> s:=s+A_0[la,i]*A_0[la,j]*EV_2[i,j];
> od;
> od;
> E_cor2[la]:=s;
> od;

```

Рассчитав восприимчивости $C_I^{(2)}$, можно решать систему уравнений (7) на поправки $a_i^{(1)}(I)$ к коэффициентам (5). Однако, определитель системы (7) в точности совпадает с определителем системы однородных уравнений (6), и следовательно, обращается в нуль. Поэтому в дополнение к (7) следует использовать условие сохранения нормировки волновой функции в первом порядке по V_D :

$$\sum_{l=m+p}^{l_{\max}} a_l^{(0)}(I) a_l^{(1)}(I) = 0 \quad (9)$$

Значения восприимчивостей $c_l^{(2)}$ вместе с коэффициентами $a_l^{(1)}(I)$ рассчитываются для каждого собственного вектора диамагнитной матрицы EV_1, соответствующего конкретному собственному значению $c_l^{(1)}$. Величины $a_l^{(1)}(I)$ определяют поправки первого порядка по B^2 к волновой функции вырожденного состояния, связанные с изменением в сильном поле вклада базисных векторов j_{nlm} .

```

Mv_1:=matrix(k,k-1,[]);
Mv_3:=matrix(k-1,k-1,[]);
Mv_4:=matrix(k-1,k-1,[]);
Dv_3:=vector(k-1,[]);
Kv_3:=vector(k-1,[]);
A_1:=vector(k,[]);
Cv_1:=matrix(k,k,[]);
Cv_2:=matrix(k,k,[]);
> for la from 1 to k do
> Cv_1:=evalm(-E_cor1[la]*Matrix(k,k,shape=identity)+EV_1):
Cv_2:=evalm(-E_cor2[la]*Matrix(k,k,shape=identity)+EV_2):
for i from 1 to k do for j from 1 to k-1 do
Mv_1[i,j]:=Cv_1[i,j]-A_0[la,j]/A_0[la,k]*Cv_1[i,k]
od od;
> Mv_3:=delrows(Mv_1,k..k);
for u from 1 to k-1 do
Dv_3[u]:=sum(A_0[la,v]*Cv_2[u,v],v=1..k);
od;
Mv_4:=concat(Dv_3,evalm(delcols(Mv_3,1..1)));
A_1[la]:=det(Mv_4)/det(Mv_3)/A_0[la,1]/3;
> od;
> print(E_cor2,A_1);

```

Таким образом, $E_cor2[la]=c_l^{(2)}$, $a_{l=1}^{(1)}(I) = A_1[la]$.

Общая поправка к матричному элементу первого порядка содержит пять слагаемых. Помимо вычисленных выше поправок к коэффициентам разложения волновых функций по сферическому кулоновскому базису (q5) следует учитывать изменение в магнитном поле матричного элемента дипольного перехода (q1 и q2), а также появление магнитоиндуцированных переходов из nf – состояний (q3 и q4). В результате, общий фактор $q_{10}^{(1)}(I)$ соответствует сумме этих поправок, формулы для вычисления которых приведены в [13]. Фрагмент программы для расчета этих компонент для $m=0$ имеет следующий вид:

```

> # Magnetic field induced corrections
> q1:=(119*n^6-277*n^4-171*n^2-55)/120/(n^2-1)^3;
> q2:=n^4*(15*n^8-80*n^6+152*n^4-84*n^2+189)/60/(n^2-1)^3;

```



```

> q3:=(1a)->
> sqrt(3)/20/sqrt(7)*A_0[1a,2]/A_0[1a,1]*2*n^2*(49*n^2-
1)*sqrt((n^2-4)*(n^2-9))/3/(n^2-1)^3;
> q4:=(1a)->
> sqrt(3)/20/sqrt(7)*A_0[1a,2]/A_0[1a,1]*n^4*sqrt((n^2-
4)*(n^2-9))/24/(n^2-1)^3*(55*n^6+115*n^4-411*n^2-527);

```

Составим из фактора $q_{10}^{(1)}$ и его компонент специальную матрицу QuB, содержащую k строк, причем каждая строка этой матрицы содержит информацию о поправках для некоторого диамагнитного состояния. Для идентификации диамагнитных состояний целесообразно ввести в матрицу QuB значения диамагнитных восприимчивостей $c_I^{(1)}$ и $c_I^{(2)}$.

```

> QuB:=array(1..k,1..8,[]);
> for lu from 1 to k do
> QuB[lu,7]:=E_cor1[lu]/8 ;
> QuB[lu,8]:=E_cor2[lu]/24;
> QuB[lu,1]:=q1;
> QuB[lu,2]:=q2;
> QuB[lu,3]:=q3(lu);
> QuB[lu,4]:=q4(lu);
> QuB[lu,5]:=A_1[lu];
> QuB[lu,6]:=sum(QuB[lu,c],c=1..5);
> od;

```

Вычисления на этом можно считать завершенными. Однако нумерация диамагнитных состояний производится программой произвольно, так как собственные значения матрицы возмущения, вычисленные процедурой `eigenvects`, получаются неупорядоченными. Для исправления этого неудобства можно осуществить упорядочение строк матрицы QuB по значению диамагнитной восприимчивости первого порядка $c_I^{(1)}$:

```

> # Сортировка матрицы результатов
> for u1 from 1 to k-1 do
> for u2 from u1+1 to k do
> if QuB[u1,7]>QuB[u2,7] then
> for u3 from 1 to 8 do
> Lo:=QuB[u1,u3];
> QuB[u1,u3]:=QuB[u2,u3];
> QuB[u2,u3]:=Lo;
> od ; fi; od;od;

```

Полученные результаты можно вывести как в виде таблицы

```

> for lur from 1 to k do
> print(lur, `
`,QuB[lur,7],`
`,QuB[lur,8],`
`,evalf(QuB[lur,6]));

```

```
> printf(`%14g      %+14g      %+14g      %+14g      %+14g`
`,evalf(QuB[lur,1]),evalf(QuB[lur,2]),evalf(QuB[lur,3]),eva
lf(QuB[lur,4]),evalf(QuB[lur,5]));
> od;
```

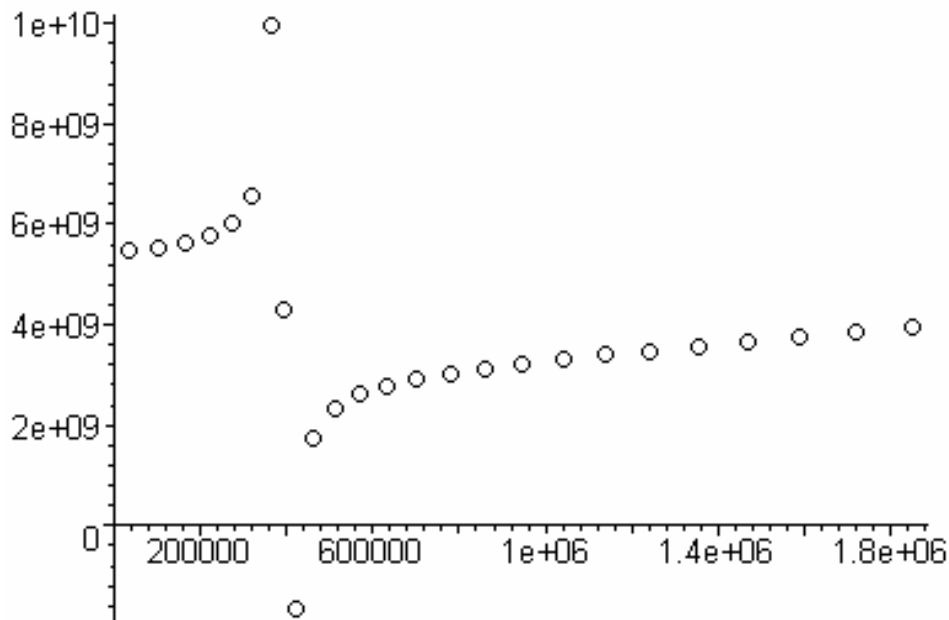
или в виде графика.

```
> plot([QuB[kisa,7],QuB[kisa,6],kisa=1..k]);
```

На рисунке 14 в качестве примера изображен график, соответствующий случаю $n=50$.

```
> # The end !
```

Рисунок 14



Аналогичным образом проводятся расчеты и для других типов переходов из вырожденного состояния в невырожденное, в частности, для всех переходов серий Лаймана и Бальмера.

Библиография

- [1] Дьяконов В.П. MAPLE 7 Учебный курс, СПб, Питер, 2002
- [2] Дьяконов В.П. MAPLE 8 в математике, физике и образовании, М. СОЛОН_Пресс, 2003, 656 с.
- [3] Аладьев В.З. Эффективная работа в MAPLE 6/7, М., ЛБЗ, 2002, 334 с.
- [4] Аладьев В.З., Лиопо В.А., Никитин А.В., Математический пакет MAPLE в физическом моделировании, Гродно 2002, 418 с.
- [5] Голоскоков Д.П. Уравнения математической физики Решение задач в системе MAPLE, СПб, Питер, 2004
- [6] Матросов А., MAPLE 6 Решение задач высшей математики и механики, СПб, BHV, 2001

- [7] Флюгге З. Задачи по квантовой механике, том 1, М. МИР, 1974, 340 с.
- [8] Варшалович Д.А., Москалев А.Н., Херсонский В.К., Квантовая теория углового момента. Л: Наука, 1975, - 440 с.
- [9] Запрыгаев С.А., Копытин И.В., Введение в алгебру угловых моментов, - Воронеж: Издательство Воронежского университета, 1980, 124 с.
- [10] Копытин И.В., Корнев А.С. Введение в алгебру угловых моментов, Учебное пособие для студентов 4-5 курсов дневного отделения физического факультета специальности 010400, Воронеж, 2002
- [11] Г. Бейтмен, А.Эрдейи, Высшие трансцендентные функции, Наука, Москва (1973) Т.1, Гл. 5; (1974), Т.2, Гл. 10.
- [12] Л. Д. Ландау и Е. М. Лифшиц, Квантовая механика, т.3, Нерелятивистская теория, Москва « Наука », (1989), 767с
- [13] В.Д. Овсянников В.В. Чернушкин, Радиационные свойства диамагнитных уровней в атомах: зависимость вероятности переходов от напряженности магнитного поля (ЖЭТФ , 1999 , том 116 выпуск 3(9) , стр. 1161 – 1183)
- [14] V.D.Ovsiannikov and S.V.Gossev, Diamagnetic shift and splitting of Rydberg levels in atoms. *Physica Scripta* **57**, 506 (1998)..

Составитель: Чернушкин Вячеслав Вячеславович
Редактор: Тихомирова О. А.