Реализация на ЭВМ алгоритма вычисления нелокальных симметрий для уравнений типа Дирака

В.И. ФУЩИЧ, В.В. КОРНЯК

Нелиевский алгоритм вычисления алгебр инвариантности систем дифференциальных уравнений в частных производных реализован в виде программ для ЭВМ. Программа написана на языке символьных вычислений PL/I–FORMAC. На ЭВМ вычислена новая восьмимерная алгебра инвариантности для системы уравнений Дирака.

The non-Lie algorithm of calculation of invariance algebras of systems of partial differential equations is realized on computer. The program is written in the symbol language PL/I–FORMAC and the new 8-dimensional invariance algebra of Dirac equation is computed.

В [1–3] предложен новый метод исследования теоретико-алгебраических свойств систем дифференциальных уравнений в частных производных, позволяющий устанавливать симметрии, которые не могут быть в принципе вычислены с помощью классического метода С. Ли. Этот метод дал возможность обнаружить неизвестные ранее нелокальные симметрии основных уравнений релятивистской квантовой теории [1–4]: Максвелла, Дирака, Кеммера–Деффина, Рариты–Швингера и т.д.

Применение нелиевского алгоритма [1–3] к конкретным системам уравнений требует, как известно, проведения большого объема аналитических вычислений, для выполнения которых целесообразно использовать ЭВМ. Эта статья посвящена реализации алгоритма [1–3] в виде программ для ЭВМ. Программа написана на языке символьных вычислений PL/I–POKMAC [5].

1. Описание алгоритмов

Ключевая идея нелиевского алгоритма [1–3] для вычисления алгебры инвариантности систем дифференциальных уравнений (ДУ) состоит в приведении их, с помощью невырожденного преобразования, к максимально расщепленным независимым подсистемам. После такого расщепления обычно бывает нетрудно найти алгебру инвариантности преобразований Фурье, задачу можно свести к проблеме приведения символьной матрицы к каноническому жорданову виду.

Приведение матрицы к жорданову виду, включает в себя в виде подзадачи проблему собственных значений, является одной из наиболее технически трудных задач линейной алгебры. Для численных матриц только в недавнее время были разработаны так называемые QR-алгоритмы [6], признанные удовлетворительными. Эти алгоритмы являются итерационными и поэтому совершенно непригодны для символьных матриц, для которых отсутствует понятие сходимости. Поэтому мы использовали более прямые методы. Алгоритм состоит из двух основных частей и реализован в виде двух отдельных программ на языке FORMAC. Первая часть

Препринт 85.20, Киев, Институт математики АН УССР, 1985, 18 с.

алгоритма служит для получения характеристических полиномов матрицы и в тех случаях, когда степень полинома не превышает четырех — вычисляются корни. Мы воспользовались здесь методом Данилевского [7] приведения матрицы к канонической форме Фробениуса:

$$F = \begin{pmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} & -\alpha_n \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Характеристический полином матрицы Фробениуса равен

$$f(\lambda) = \lambda^n + \alpha_1 \lambda^{n-1} + \cdots + \alpha_n,$$

а любая матрица приводится к форме Фробениуса (точнее к блочно-треугольной форме с фробениусовыми матрицами на диагонали) с помощью элементарных преобразований, не требующих знаний собственных значений.

Пусть имеется матрица

$$A = A^0 = \begin{pmatrix} a^0_{11} & a^0_{12} & \cdots & a^0_{1,n-1} & a^0_{1,n} \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ a^0_{n,1} & a^0_{n,2} & \cdots & a^0_{n,n-1} & a^0_{n,n} \end{pmatrix}.$$

Здесь и ниже верхний индекс нумерует последовательность матриц, возникающих в процессе вычислений.

Алгоритм приведения включает следующие действия:

1) Начинаем с предпоследнего элемента нижней строки $a^0_{n,n-1}$. Если этот элемент не равен нулю — делим на него все элементы столбца, а из остальных столбцов вычитаем $(n-1)$-й столбец, умноженный на такие величины, чтобы в нижней строке получались нули. В результате получаем матрицу вида

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1,n-1} & b_{1,n} \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

Эта матрица не подобна матрице A^0. Чтобы восстановить подобие, нужно еще выполнить некоторое, также элементарное, преобразование, эквивалентное умножению слева на матрицу

$$M = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ a^0_{n,1} & a^0_{n,2} & \cdots & a^0_{n,n-1} & a^0_{n,n} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Это преобразование изменяет лишь $(n-1)$-ю строку матрицы B. Возникшая матрица

$$A^1 = MB = \begin{pmatrix} a^1_{11} & a^1_{12} & \cdots & a^1_{1,n-1} & a^1_{1,n} \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

уже будет подобна матрице A^0. После этого переходим к $(n-1)$-й строке и повторяем операцию с элементом, расположенным в позиции $(n-1, n-2)$.

2) Если очередной элемент в позиции $(n-k, n-k-1)$ оказался равным нулю — ищем слева от него в $(n-k)$-й строке ненулевой элемент. Допустим, что он расположен в l-м столбце, — переставляем $(n-k-1)$-й и l-й столбец и одновременно, для сохранения подобия, $(n-k-1)$-ю и l-ю строки.

3) Если ненулевой элемент в $(n-k)$-й строке не найден, то матрица A^k приобретает блочно-треугольный вид

$$A^k = \left(\begin{array}{c|c} C & D \\ \hline 0 & F \end{array}\right),$$

где

$$F = \begin{pmatrix} a^k_{n-k,n-k} & \cdots & a^k_{n-k,n-1} & a^k_{n-k,n} \\ 1 & \cdots & 0 & 0 \\ \cdot & \cdots & \cdot & \cdot \\ 0 & \cdots & 1 & 0 \end{pmatrix},$$

т.е. F имеет фробениусову форму.

Поскольку

$$\det(A - \lambda E) = \det(A_k - \lambda E) = \det(C - \lambda E)\det(F - \lambda E),$$

сразу можно выделить $\det(F - \lambda E)$ как сомножитель характеристического полинома и далее применить алгоритм к матрице C.

В соответствии с этим алгоритмом была написана программа NLP. Эта программа вычисляет и корни полиномов, если их степень оказывается меньше пяти. Для более экономного представления разреженных матриц в памяти ЭВМ мы применяем представление строк матрицы в виде сумм, слагаемыми которых является произведения элементов матрицы на координаты "пробного" вектора. Ниже будет приведен поясняющий пример.

После определения собственных чисел матрицы A и их алгебраических кратностей необходимо найти жорданову форму J и матрицу преобразования W, такую, что

$$WAW^{-1} = J.$$

Для этих целей была написана программа NLJW. Алгоритм работает следующим образом:

Выполняется цикл последовательного перебора корней. Выбрав определенный корень, мы должны установить структуру жорданова блока, соответствующего этому корню. Жордановым блоком мы здесь будем называть совокупность всех жордановых клеток с данным корнем λ. Жордановой клеткой порядка p называют матрицу размера $p \times p$ вида

$$\begin{pmatrix} \lambda & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & \lambda & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda \end{pmatrix}.$$

В частности, диагональному случаю соответствуют жордановы клетки порядка 1. Существует общая формула, позволяющая найти число n_p жордановых клеток порядка p в блоке с собственным числом λ, именно

$$n_p = \operatorname{rank}(A-\lambda E)^{p-1} - 2\operatorname{rank}(A-\lambda E)^{p} + \operatorname{rank}(A-\lambda E)^{p+1}.$$

По определению полагается $\operatorname{rank}(A-\lambda E)^0 = N$, где N — размер матрицы A. Непосредственное использование этой формулы может привести к излишним вычислениям. Во многих случаях (на практике — в большинстве) для определения числа n_p нет необходимости вычислять $\operatorname{rank}(A-\lambda E)^{p+1}$. Вычисление высоких степеней матрицы $A-\lambda E$ является самым узким местом алгоритма. Например, если $A-\lambda E$ — разреженная матрица, то уже $(A-\lambda E)^2$ как правило не будет разреженной. Поэтому важно минимизировать количество вычислений степеней матриц. Мы получим некоторые соотношения, позволяющие достичь этого.

Поскольку все рассматриваемые ниже величины не зависят от выбора системы координат, рассмотрим матрицу $A-\lambda E$ в жордановом базисе. Она будет иметь вид

$$A-\lambda E = \left(\begin{array}{cccc} \left.\begin{array}{ccc}\boxed{0}&&\\&\ddots&\\&&\boxed{0}\end{array}\right\}n_1 &&& \\ & \left.\begin{array}{ccc}\boxed{\begin{matrix}0&1\\0&0\end{matrix}}&&\\&\ddots&\\&&\boxed{\begin{matrix}0&1\\0&0\end{matrix}}\end{array}\right\}n_2 && \\ && \ddots & \\ &&& \boxed{B} \end{array}\right),$$

здесь B — матрица полного ранга $\operatorname{rank} B = N - k_{\mathrm{alg}}$, $k_{\mathrm{alg}} = n_1 + 2n_2 + \cdots + mn_m$ — алгебраическая кратность корня λ. Последовательно возведение в степень нильпотентной матрицы вида

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix}.$$

приводит к сдвигам на 1 позицию вправо диагонали из единиц и уменьшению ранга на 1. Исходя из этого, рассматривая последовательно ранги степеней матрицы $A-\lambda E$ можно получить необходимые нам величины. Допустим, что уже найдены числа $n_1, n_2, \ldots, n_{p-1}$ и вычислен $\operatorname{rank}(A-\lambda E)^p$, тогда нам известны величины

$$\begin{aligned} a &= pn_p + (p+1)n_{p+1} + \cdots + mn_m = k_{\mathrm{alg}} - [n_1 + 2n_2 + \cdots + (p-1)n_{p-1}], \\ b &= n_{p+1} + 2n_{p+2} + \cdots + (m-p)n_m = k_{\mathrm{alg}} - N + \operatorname{rank}(A-\lambda E)^p. \end{aligned}$$

Поскольку a, b, n_l — неотрицательные числа, то в случаях когда $b = 0$ или $b = 1$ величины $n_p, n_{p+1}, \ldots$ определяются однозначно. Именно, если $b = 0$, то

$n_p = a/p$, $n_{p+1} = 0, \ldots$; если $b = 1$, то $n_p = (a - p - 1)/p$, $n_{p+1} = 0, \ldots$ (Можно объединить формулу вычисления n_p для обоих случаев $b = 0$ и $b = 1$: $n_p = [a - b(p + 1)]/p$). При $p = 1$, например, случай $b = 0$ соответствует равенству геометрической кратности алгебраической, что приводит к диагонализации блока, а равенство $b = 1$ означает, что геометрическая кратность на единицу меньше алгебраической, что указывает на наличие в жордановом блоке диагональной части длины $k_{\mathrm{alg}} - 2$ и одной жордановой клетки вида

$$\begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}.$$

Таким образом, в подобных случаях для определения n_p достаточно знать только $\operatorname{rank}(A - \lambda E)^p$. Отдельно необходимо выделить случай алгебраической кратности равной единице — этот случай вообще не требует вычисления рангов.

В процессе определения жордановой структуры матрицы можно по частям решать матричное уравнение $WA - JW = 0$ для матрицы преобразования W. Именно, можно показать, что каждой жордановой метке соответствует автономная система линейных уравнений, содержащая только те элементы матрицы W, которые входят в строки W, расположенные против соответствующих строк жордановой клетки. Более того, если в блоке имеется несколько жордановых клеток одного размера, то им соответствуют системы уравнений одинаковой структуры. Поэтому достаточно решать систему уравнений один раз и, заменив переменные, получить для всех клеток данного размера решения. Проиллюстрируем это примером для матриц размера 2×2. Пусть матрица J имеет вид

$$J = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix},$$

тогда уравнение $WA - JW = 0$ будет иметь вид

$$\begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} = 0.$$

Для верхней строки матрицы W получаем систему уравнений:

$$\begin{aligned} &w_{11}(a_{11} - \lambda) + w_{12}a_{21} = 0, \\ &w_{11}a_{12} + w_{12}(a_{22} - \lambda) = 0, \end{aligned}$$

для нижней — систему

$$\begin{aligned} &w_{21}(a_{11} - \lambda) + w_{22}a_{21} = 0, \\ &w_{21}a_{12} + w_{22}(a_{22} - \lambda) = 0. \end{aligned}$$

Мы видим, что эти системы одинаковы с точностью до замены неизвестных. Аналогичная ситуация и в случае жордановых клеток общего вида.

Таким образом алгоритм производит следующие действия:

1) Выполняется цикл по различным корням.

2) После выбора корня выполняется цикл по размерам жордановых клеток. Для каждого размера определяется количество клеток, и, если оно не равно нулю — методом Гаусса решается уравнение для определения соответствующих элементов матрицы свободными параметрами, а остальные выражаются через них.

3) Последовательно выводятся на печать по клеткам строки матриц J и W. Производится подсчет числа произвольных параметров.

Такая последовательная организация вычислений может оказаться полезной в случае, если известны не все корни. В таком случае мы получим частичное приведение матрицы к жорданову виду, что может дать некоторую информацию о симметриях системы уравнений.

В конце работы программа печатает число параметров матрицы преобразования. Параметры должны удовлетворять условиям невырожденности матрицы. Невырожденность матрицы соответствует случаю общего положения, т.е. из любой вырожденной матрицы малым изменением параметров можно получить невырожденную.

Мы не вычисляем здесь обратную матрицу W^{-1} по следующим причинам:

1) Элементы обратной матрицы будут слишком громоздкими, т.к. они представляют собой сложные нелинейные функции параметров.

2) Даже в численной математике непосредственное вычисление любого объекта, получаемого с помощью обратных матриц требует в 3 раза меньше времени и в 2 раза меньше памяти, чем обращение матрицы. Поэтому с вычислительной точки зрения обратные матрицы бесполезны.

Для получения выражений, содержащих обратные матрицы целесообразней написать программу, вычисляющую эти выражения непосредственно, например, методом Гаусса.

2. Примеры применения программ

Приведем примеры, иллюстрирующие работу программ NLP и NLJW. Все вычисления проводились на ЭВМ ЕС–1022 с доступной оперативной памятью 390 килобайтов.

Рассмотрим вначале систему уравнений с минимальным нетривиальным размером матрицы:

$$\begin{aligned}&\frac{\partial u}{\partial t}-\frac{\partial v}{\partial x}=0,\\&-\frac{\partial u}{\partial x}+\frac{\partial v}{\partial t}=0.\end{aligned}$$

Эта система получается в результате факторизации волнового уравнения

$$\frac{\partial^2 u}{\partial t^2}-\frac{\partial^2 u}{\partial x^2}=0.$$

У этой системы нет нетривиальных нелокальных симметрий рассматриваемого нами типа. Мы выбрали ее только из-за краткости ввода и вывода. Символ оператора системы имеет вид

$$\begin{pmatrix} p_0 & p_1\\ p_1 & p_0\end{pmatrix},\qquad \text{где}\quad p_0=i\frac{\partial}{\partial t},\quad p_1=-i\frac{\partial}{\partial x}.$$

Ниже слева приведена информация в том виде, в котором она представлена на вводе и выводе — справа — эквивалент в обычных математических обозначениях или пояснение.

Для работы программы NLP необходимо ввести следующую информацию:

```
СИСТЕМА UT - VX =0, -UX - VT=0
                  — пояснительная информация.
'P0*#1+P1*#2'     — в таком виде вводится матрица. Здесь
                  “пробный” вектор с координатами #1, #2
'P1*#1+P0*#2'     служит для представления строк матрицы
                  в виде сумм.
```

На выходе будет напечатано:

```
ВХОДНЫЕ ДАННЫЕ
СИСТЕМА UT - VX =0, -UX - VT=0
МАТРИЦА
   P0*#1+P1*#2
   P1*#1+P0*#2
ХАРАКТЕРИСТИЧЕСКИЙ ПОЛИНОМ
CHP(1)=-2L#P0+L#^2+P0^2-P1^2,  L# — обозначение для
                               собственного числа.
КОРНИ
   L#=P0+P1
   L#=P0-P1
```

Для работы программы NLJW нужно ввести следующую информацию:

```
СИСТЕМА UT - VX =0, -UX - VT=0
2                 — число различных корней
'P0+P1'   1       — корни и их алгебраические
'P0-P1'   1       кратности
'P0*#1+P1*#2'     – исходная матрица
'P1*#1+P0*#2'
```

На выходе будет напечатано:

```
ВХОДНЫЕ ДАННЫЕ
СИСТЕМА UT - VX =0, -UX - VT=0
ЧИСЛО РАЗЛИЧНЫХ КОРНЕЙ 2
СОБСТВЕННЫЕ ЧИСЛА И ИХ КРАТНОСТЬ
   P0+P1   1
   P0-P1   1
СТРОКИ ИСХОДНОЙ МАТРИЦЫ
   P0*#1+P1*#2
   P1*#1+P0*#2
ПЕЧАТЬ СТРОК МАТРИЦЫ ПО ЖОРДАНОВЫМ КЛЕТКАМ
КОРЕНЬ АЛГЕБРАИЧЕСКИЙ КРАТНОСТИ 1
   L#(1)=P0+P1
ЖОРДАНОВА КЛЕТКА ПОРЯДКА 1
   J#(1)=#1(P0+P1)      — первая строка (p0 + p1, 0)
                        жордановой матрицы J.
СОТВЕТСТВУЯЩАЯ ЧАСТЬ МАТРИЦЫ ПРЕОБРАЗОВАНИЯ
   W#(1)=T#1#1+T#1#2    — первая строка (t1, t1)
                        матрицы W. Здесь T#1 —
                        обозначение для произвольного
                        параметра t1.
КОРЕНЬ АЛГЕБРАИЧЕСКОЙ КРАТНОСТИ 1
   L#(1)=P0-P1
```

```
ЖОРДАНОВА КЛЕТКА ПОРЯДКА 1
    J#(2)=#2(P0-P1)
```
— вторая строка $(0, p_0 - p_1)$ матрицы J.

```
СООТВЕТСТВУЮЩАЯ ЧАСТЬ МАТРИЦЫ ПРЕОБРАЗОВАНИЯ
    W#(2)=-T#2#1+T#2#2
```
— вторая строка $(-t_2, t_2)$ матрицы W.

```
МАТРИЦА ПРЕОБРАЗОВАНИЯ 2-ПАРАМЕТРИЧЕСКАЯ
```

Приведем результаты применения программ к уравнению Дирака и уравнению Кеммера–Дефина–Петье в 4-мерном пространстве–времени для 10-компонентной волновой функции.

Символ оператора Дирака имеет вид

$$\begin{pmatrix} p_0 - m & 0 & -p_3 & -p_1 + ip_2 \\ 0 & p_0 - m & -p_1 - ip_2 & p_3 \\ p_3 & p_1 - ip_2 & -p_0 - m & 0 \\ p_1 + ip_2 & -p_3 & 0 & -p_0 - m \end{pmatrix}.$$

С помощью программы NLP получаем 2 корня

$$\lambda_{1,2} = -m \pm \mathcal{P}, \qquad \mathcal{P} = \sqrt{p_\mu p^\mu}.$$

Каждый из этих корней имеет алгебраическую кратность равную двум.

В результате работы программы NLJW получаем диагональную жорданову матрицу

$$J = \begin{pmatrix} -m + \mathcal{P} & & & \\ & -m + \mathcal{P} & & \\ & & -m + \mathcal{P} & \\ & & & -m + \mathcal{P} \end{pmatrix}$$

и 8-параметрическую матрицу преобразования

$$W = \begin{pmatrix} \dfrac{-t_1 p_3 - t_2(p_1 + ip_2)}{p_0 - \mathcal{P}} & \dfrac{-t_1(p_1 - ip_2) + t_2 p_3}{p_0 - \mathcal{P}} & t_1 & t_2 \\ \dfrac{-t_3 p_3 - t_4(p_1 + ip_2)}{p_0 - \mathcal{P}} & \dfrac{-t_3(p_1 - ip_2) + t_4 p_3}{p_0 - \mathcal{P}} & t_3 & t_4 \\ \dfrac{-t_5 p_3 - t_6(p_1 + ip_2)}{p_0 + \mathcal{P}} & \dfrac{-t_5(p_1 - ip_2) + t_6 p_3}{p_0 + \mathcal{P}} & t_5 & t_6 \\ \dfrac{-t_7 p_3 - t_8(p_1 + ip_2)}{p_0 + \mathcal{P}} & \dfrac{-t_7(p_1 - ip_2) + t_8 p_3}{p_0 + \mathcal{P}} & t_7 & t_8 \end{pmatrix}. \qquad (1)$$

Теперь видно, что символ оператора Дирака инвариантен по отношению к линейным преобразованиям из 8-мерой группы $GL(2) \otimes GL(2)$. При переходе с помощью преобразования Фурье от символов к операторам, матрицы групп инвариантности станут интегродифференциальными операторами.

Непосредственное обращение матрицы W можно выполнить с помощью ЭВМ, но это приводит к громоздким и плохо отражающим структуру выражениям. Поэтому лучше воспользоваться факторизованным представлением матрицыW, предложенным в [3]

$$W = GW_0. \qquad (2)$$

Здесь W_0 — значение матрицы (1) при каком-нибудь фиксированном наборе параметров t_1—t_8, G — произвольная матрица из $GL(2)\otimes GL(2)$. При таком представлении обратная матрица выражается формулой

$$W^{-1} = W_0^{-1}G^{-1}.$$

При выборе конкретных значений параметров t_1—t_8 потребуем, чтобы матрица W_0 не вырождалась при обращении в нуль пространственных компонент p_1, p_2, p_3 4-вектора импульса. Легко найти набор параметров, удовлетворяющих этому требованию. Например, можно выбрать:

$$t_1 = -\frac{p_3}{p_0+\mathcal{P}}, \qquad t_2 = -\frac{p_1-ip_2}{p_0+\mathcal{P}}, \qquad t_3 = -\frac{p_1+ip_2}{p_0+\mathcal{P}},$$
$$t_4 = \frac{p_3}{p_0+\mathcal{P}}, \qquad t_5 = t_8 = 1, \qquad t_6 = t_7 = 0.$$

При этих значениях параметров матрица (1) примет вид

$$W_0 = \begin{pmatrix} 1 & 0 & -\dfrac{p_3}{p_0+\mathcal{P}} & -\dfrac{p_1-ip_2}{p_0+\mathcal{P}} \\ 0 & 1 & -\dfrac{p_1+ip_2}{p_0+\mathcal{P}} & \dfrac{p_3}{p_0+\mathcal{P}} \\ -\dfrac{p_3}{p_0+\mathcal{P}} & -\dfrac{p_1-ip_2}{p_0+\mathcal{P}} & 1 & 0 \\ -\dfrac{p_1+ip_2}{p_0+\mathcal{P}} & \dfrac{p_3}{p_0+\mathcal{P}} & 0 & 1 \end{pmatrix} = 1 - \frac{\gamma_0\gamma_i p_i}{p_0+\mathcal{P}}.$$

Матрицу, обратную к W_0, естественно искать в виде

$$W_0^{-1} = a\left(1 + \frac{\gamma_0\gamma_i p_i}{p_0+\mathcal{P}}\right).$$

Коэффициент a находится из соотношения

$$W_0W_0^{-1} = a\left\{1 - \frac{\gamma_0\gamma_i p_i\gamma_0\gamma_j p_j}{(p_0+\mathcal{P})^2}\right\} = a\left\{1 - \frac{p_1^2+p_2^2+p_3^2}{(p_0+\mathcal{P})^2}\right\} = 1.$$

Таким образом

$$a = 1 + \frac{p_1^2+p_2^2+p_3^2}{(p_0+\mathcal{P})^2 - p_1^2 - p_2^2 - p_3^2}.$$

Если матрица G имеет следующую параметризацию

$$G = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & b_{11} & b_{12} \\ 0 & 0 & b_{12} & b_{22} \end{pmatrix},$$

то обратная к ней матрица примет вид

$$G^{-1} = \begin{pmatrix} \dfrac{1}{\Delta_1}\begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix} & \\ & \dfrac{1}{\Delta_2}\begin{pmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{pmatrix} \end{pmatrix}.$$

Здесь a_{ij}, b_{ij} —произвольные параметры, удовлетворяющие условиям

$$\Delta_1 = a_{11}a_{22} - a_{21}a_{12} \neq 0,$$
$$\Delta_2 = b_{11}b_{22} - b_{21}b_{12} \neq 0.$$

Рассмотрим теперь символ оператора Гамильтона для уравнения Дирака

$$\gamma_0\gamma_i p_i + \gamma_0 m = \begin{pmatrix} m & 0 & p_3 & p_1 - ip_2 \\ 0 & m & p_1 + ip_2 & -p_3 \\ p_3 & p_1 - ip_2 & -m & 0 \\ p_1 + ip_2 & -p_3 & 0 & -m \end{pmatrix}.$$

В результате применения вычислительных программ получаем жорданову форму

$$J = \begin{pmatrix} E & & 0 & \\ & E & & \\ & & -E & \\ & 0 & & -E \end{pmatrix}$$

и 8-параметрискую матрицу преобразования

$$W = \begin{pmatrix} (m+E)[t_1p_3 + t_2(p_1+ip_2) & (m+E)[t_1(p_1-ip_2) - t_2p_3] & t_1(p_1^2+p_2^2+p_3^2) & t_2(p_1^2+p_2^2+p_3^2) \\ (m+E)[t_3p_3 + t_4(p_1+ip_2) & (m+E)[t_3(p_1-ip_2) - t_4p_3] & t_3(p_1^2+p_2^2+p_3^2) & t_4(p_1^2+p_2^2+p_3^2) \\ (m-E)[t_5p_3 + t_6(p_1+ip_2) & (m-E)[t_5(p_1-ip_2) - t_6p_3] & t_5(p_1^2+p_2^2+p_3^2) & t_6(p_1^2+p_2^2+p_3^2) \\ (m-E)[t_7p_3 + t_8(p_1+ip_2) & (m-E)[t_7(p_1-ip_2) - t_8p_3] & t_7(p_1^2+p_2^2+p_3^2) & t_8(p_1^2+p_2^2+p_3^2) \end{pmatrix}. \quad (3)$$

Здесь $E = \sqrt{p_1^2 + p_2^2 + p_3^2 + m}$.

Выбирая для матрицы (3) набор параметров аналогично предыдущему случаю

$$t_1 = \frac{p_3}{p_0 + E}, \qquad t_2 = \frac{p_1 - ip_2}{p_0 + E}, \qquad t_3 = \frac{p_1 + ip_2}{p_0 + E},$$
$$t_4 = -\frac{p_3}{p_0 + E}, \qquad t_5 = t_8 = 1, \qquad t_6 = t_7 = 0$$

получим матрицу

$$W_0 = \begin{pmatrix} 1 & 0 & \dfrac{p_3}{m+E} & \dfrac{p_1 - ip_2}{m+E} \\ 0 & 1 & \dfrac{p_1 + ip_2}{m+E} & -\dfrac{p_3}{m+E} \\ -\dfrac{p_3}{m+E} & -\dfrac{p_1 - ip_2}{m+E} & 1 & 0 \\ -\dfrac{p_1 + ip_2}{m+E} & \dfrac{p_3}{m+E} & 0 & 1 \end{pmatrix}.$$

Матрица, обратная к W_0, имеет вид

$$W_0^{-1} = \left\{1 - \frac{p_1^2 + p_2^2 + p_3^2}{(m+E)^2 + p_1^2 + p_2^2 + p_3^2}\right\} \left(1 - \frac{\gamma_i p_i}{m+E}\right).$$

Используя формулу (2) получим общий вид обратной матрицы к матрице преобразования (3).

Рассмотрим теперь уравнение Кеммера–Деффина–Петье

$$(\beta_\mu p^\mu - m)\Psi = 0, \qquad \mu = \overline{0,3}.$$

Матрицы β_μ удовлетворяют алгебре

$$\beta_\mu\beta_\nu\beta_\lambda + \beta_\lambda\beta_\nu\beta_\mu = g_{\mu\nu}\beta_\lambda + g_{\nu\lambda}\beta_\mu,$$

$g_{\mu\nu} = \mathrm{diag}\,(1,-1,-1,-1)$ — метрический тензор.

Матрицы β_μ можно реализовать в виде

$$\beta_0 = i\begin{pmatrix} \hat{0} & \hat{0} & -\hat{1} & 0 \\ \hat{0} & \hat{0} & \hat{0} & 0 \\ \hat{1} & \hat{0} & \hat{0} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \beta_a = \begin{pmatrix} \hat{0} & \hat{0} & \hat{0} & \lambda_a \\ \hat{0} & \hat{0} & -\hat{S}_a & 0 \\ \hat{0} & \hat{S}_a & \hat{0} & 0 \\ -\lambda_a^+ & 0 & 0 & 0 \end{pmatrix}, \ a = \overline{1,3}.$$

Здесь

$$\hat{S}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \quad \hat{S}_2 = \begin{pmatrix} 0 & 0 & i \\ 0 & 0 & 0 \\ -i & 0 & 0 \end{pmatrix}, \quad \hat{S}_3 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\lambda_1 = \begin{pmatrix} i \\ 0 \\ 0 \end{pmatrix}, \quad \lambda_2 = \begin{pmatrix} 0 \\ i \\ 0 \end{pmatrix}, \quad \lambda_1 = \begin{pmatrix} 0 \\ 0 \\ i \end{pmatrix},$$

$\hat{0}$ и $\hat{1}$ — нулевые и единичные матрицы размера 3×3, 0 — 3-х компонентные нулевые столбцы или строки.

Применение программы NLP к символу оператора КДП дает следующие корни и алгебраические кратности:

$\lambda_1 = -m$ — кратность $=4$,

$\lambda_2 = -m + \mathcal{P}$ — кратность $=3$,

$\lambda_3 = -m - \mathcal{P}$ — кратность $=3$.

С помощью программы NLJW символ оператора КДП приводится к диагональному виду, т.е. алгебраические кратности совпадают с геометрическими. При этом вычисляется 34-параметрическая матрица преобразования W, которую мы для экономии места не приводим. Таким образом уравнение Кеммера–Деффина–Петье инвариантно относительно 34-параметрической группы $GL(4)\otimes GL(3)\otimes GL(3)$.

1. Фущич В.И., О дополнительной инвариантности релятивистских уравнений движения, *Теор. и мат. физика*, 1971, **7**, № 1, 3–12.
2. Фущич В.И., О новом методе исследования групповых свойств уравнений математической физики, *ДАН СССР*, 1979, **246**, № 4, 846–850.
3. Фущич В.И., О новом методе исследования групповых свойств систем дифференциальных уравнений в частных производных, В кн.: Теоретико-групповые методы в математической физике, Киев, Ин-т математики АН УССР, 1978, 5–44.
4. Фущич В.И., Никитин А.Г., Симметрия уравнений Максвелла, Киев, Наукова думка, 1983, 200 с.
5. Tobey R., Baker J., Crews R. et al., PL/I–FORMAC symbolic mathematics interpreter, IBM, Proj. No 360-03.3.004, Contributing Program Library, IBM, 1969.
6. Воеводин В.В., Кузнецов Ю.А., Матрицы и вычисления, М., Наука, 1984, 320 с.
7. Демидович Б.П., Марон И.А., Основы вычислительной математики, М., Наука, 1970, 404 с.