

8081

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

ДУБНА



8081

ЭКЗ. ЧИТ. ЗАЛА

11 - 8081

Ф.А.Гареев, Т.П.Пузынина, И.В.Пузынин,
Р.М.Ямалеев

ПРОГРАММА ВЫЧИСЛЕНИЯ ОДНОЧАСТИЧНЫХ
ВОЛНОВЫХ ФУНКЦИЙ В ДЕФОРМИРОВАННОМ
ЯДРЕ С ПОМОЩЬЮ НЕПРЕРЫВНОГО АНАЛОГА
МЕТОДА НЬЮТОНА

1974

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ И АВТОМАТИЗАЦИИ

11 - 8081

Ф.А.Гареев , Т.П.Пузынина, И.В.Пузынин,
Р.М.Ямалеев

ПРОГРАММА ВЫЧИСЛЕНИЯ ОДНОЧАСТИЧНЫХ
ВОЛНОВЫХ ФУНКЦИЙ В ДЕФОРМИРОВАННОМ
ЯДРЕ С ПОМОЩЬЮ НЕПРЕРЫВНОГО АНАЛОГА
МЕТОДА НЬЮТОНА

В ядерной физике при описании многих физических процессов в рамках классической квантовой механики обычно сталкиваются с проблемой решения уравнения Шредингера

$$(T + V(\vec{r}) - E) \psi(\vec{r}) = 0. \quad (I)$$

В зависимости от выбора потенциала и характера исследуемой задачи уравнение (I) может принимать различные нелинейные формы. Так, при исследовании многоканальных процессов уравнение Шредингера (I) обычно приводится к задаче Штурма-Лиувилля для системы связанных дифференциальных уравнений

$$\varphi_i^{(1)} \equiv U_i''(x) + (K_{ii}(x) - \lambda) U_i(x) = \sum_{i \neq j} K_{ij}(x) U_j(x) \quad (2a)$$

на интервале $0 \leq x < \infty$ с граничными условиями

$$U_i(0) = U_i(\infty) = 0 \quad (2б)$$

(здесь и в дальнейшем индекс i принимает значения от 1 до N).

В настоящей работе предлагаются метод и алгоритм решения задачи (2) на основе непрерывного аналога метода Ньютона^{/1/}, являющиеся обобщением и развитием метода, примененного в работе^{/3/} для системы двух уравнений. Предлагаемый алгоритм реализован в написанной на ФОРТРАНе программе COSDEQ, которую можно применять непосредственно, в частности, на ЭВМ БЭСМ-6 и машинах СДС. Текст программы COSDEQ приводится в приложении.

2. Метод решения (непрерывный аналог метода Ньютона)

В излагаемом ниже подходе краевая задача (2) доопределяется добавлением условия нормировки

$$\varphi_i^{(2)} \equiv \sum_{i=1}^N \int_a^b \mathcal{U}_i^2(x) dx + \sum_{i=1}^N C_i^2 \int_a^b \mathcal{W}_i^2(x) dx - 1 = 0. \quad (2c)$$

Это условие позволяет также учитывать особенности тех задач, в которых решение задачи (2) находится на полубесконечном интервале $a \leq x < \infty$. Здесь $\mathcal{W}_i(x)$ - асимптотическое выражение при $X \rightarrow \infty$ для искомого собственного функции

$\mathcal{U}_i(x)$, которые считаются известными. Коэффициенты C_i обеспечивают в некоторой достаточно удаленной точке выполнение равенств

$$\mathcal{U}_i(b) = C_i \mathcal{W}_i(b).$$

Если задача (2) рассматривается на конечном отрезке $[a, b]$, то в условии (2c) следует положить $C_i = 0$. Для уравнения (2a) рассматриваются граничные условия:

$$\varphi_i^{(3)}(\lambda, a) \equiv \mathcal{U}_i(a) = 0; \quad (2d)$$

$$\varphi_i^{(4)}(\lambda, b) \equiv \mathcal{U}_i(b) + k_i(\lambda, b) \mathcal{U}_i(b) = 0,$$

аппроксимирующие исходные граничные условия на полубесконечном интервале.

Задача (2) является нелинейным функциональным уравнением

$$\varphi(z) = 0,$$

где φ - совокупность операторов $\{\varphi_i^{(j)}, j = 1, 2, 3, 4\}$.

определенных в (2), а $Z = (\lambda, \vec{\mathcal{U}}(x)) \in R \times C^2[a, b]$.

Для решения этого уравнения можно применить непрерывный аналог метода Ньютона [1]. Согласно этому подходу вводится непрерывный параметр t ($0 \leq t < \infty$), от которого зависят искомые величины в задаче (2) $Z(t) = (\lambda(t), \vec{\mathcal{U}}(x, t))$, а задача (2) заменяется эволюционным уравнением

$$\varphi'(Z(t)) \cdot Z(t) = -\varphi(Z(t)), \quad Z(0) = (\lambda_0, \mathcal{Y}_0(x)). \quad (3)$$

Здесь φ' - производная Фреше оператора φ , а $Z'(t) = (\mu(t), \mathcal{Z}(x, t))$, где

$$\mu(t) = \lambda'(t); \quad \mathcal{Z}(x, t) = \mathcal{Y}_t(x, t). \quad (4)$$

При достаточно общих предположениях [2], [4]

$$\lim \|Z(t) - Z^*\| = 0,$$

где $Z^* = (\lambda^*, \vec{\mathcal{U}}^*(x))$ - искомое решение задачи (2).

Дискретное представление по параметру t

Дискретную аппроксимацию эволюционного уравнения (3) по параметру t можно реализовать на основе метода Эйлера [1], [3]. Полуось $0 \leq t < \infty$ разбивается узловыми точками t_k ($k = 0, 1, 2, \dots$) на интервалы τ_k , при этом:

$$t_{k+1} = t_k + \tau_k, \quad t_0 = 0. \quad (5a)$$

В дальнейшем, для краткости в обозначении величин, зависящих от t , будем использовать индекс K при $t = t_K$, опуская запись самого аргумента.

Выражения (4) заменим разностными аналогами

$$\lambda_{K+1} = \lambda_K + \tau_K \mu_K; \quad \mathcal{U}_{i,K+1}^{(x)} = \mathcal{U}_{i,K}^{(x)} + \tau_K \mathcal{V}_{i,K}^{(x)} \quad (5b)$$

Схему приближенного решения эволюционного уравнения (3) для задачи (2) методом Эйлера можно описать следующим образом. Предположим, что при $t = t_K$ значения λ_K и $\mathcal{U}_{i,K}(x)$ известны. Тогда мы приходим к рассмотрению однопараметрического семейства краевых задач с параметрами μ_K относительно функции $\mathcal{V}_{i,K}(x)$, в которое трансформируется исходная стационарная задача (2a)-(2c):

$$\mathcal{V}_{i,K}'' + [K_{ii}(x) - \lambda_K] \mathcal{V}_{i,K} + \sum_{i \neq j} K_{ij}(x) \mathcal{V}_{j,K} = - \left\{ \mathcal{U}_{i,K}'' + [K_{ii}(x) - \lambda_K] \mathcal{U}_{i,K} + \sum_{i \neq j} K_{ij}(x) \mathcal{U}_{j,K} \right\} + \mu_K \mathcal{U}_{i,K}, \quad (6)$$

$$\mathcal{V}_{i,K}(0) = 0,$$

$$\mathcal{V}_{i,K}'(b) + f_i(\lambda, b) \mathcal{V}_{i,K}(b) = - \varphi_i^{(4)}(\lambda, b) - \mu_K \frac{\partial f_i(\lambda, b)}{\partial \lambda} \mathcal{U}_{i,K}(b).$$

Решение этой краевой задачи можно представить в виде

$$\mathcal{V}_{i,K}^{(x)}(\mu_K) = V_{i,K}(x) + \mu_K W_{i,K}(x), \quad (7)$$

где функции $V_{i,K}(x)$ и $W_{i,K}(x)$ являются решениями задачи:

$$V_{i,K}''(x) + [K_{ii}(x) - \lambda_K] V_{i,K} + \sum_{i \neq j} K_{ij}(x) V_{j,K} = - \left\{ \mathcal{U}_{i,K}'' + [K_{ii}(x) - \lambda_K] \mathcal{U}_{i,K} + \sum_{i \neq j} K_{ij}(x) \mathcal{U}_{j,K} \right\}, \quad (8A)$$

$$V_{i,K}(0) = 0, \quad V_{i,K}'(b) + V_{i,K}(b) f_i(\lambda, b) = \varphi_i^{(4)}(\lambda, b).$$

$$W_{i,K}''(x) + [K_{ii}(x) - \lambda_K] W_{i,K} + \sum_{i \neq j} K_{ij}(x) W_{j,K} = \mathcal{U}_{i,K};$$

$$W_{i,K}(0) = 0, \quad (8B)$$

$$W_{i,K}'(b) + W_{i,K}(b) f_i(\lambda, b) = \frac{\partial f_i(\lambda, b)}{\partial \lambda} \mathcal{U}_{i,K}(b). \quad (8B)$$

Решив краевые задачи (8A)-(8B), можно определить однопараметрическое семейство функций $\mathcal{V}_{i,K}^{(x)}$. Используя условия нормировки (2c) в эволюционном уравнении (3), получаем следующее выражение для определения μ_K :

$$\sum_{i=1}^N \int_a^b \mathcal{U}_{i,K} [V_{i,K} + \mu_K W_{i,K}] dx + \mu_K \sum_{i=1}^N c_{i,K}^2 \int_b^L \omega_i(\lambda, x) \frac{\partial \omega_i(\lambda, x)}{\partial \lambda} dx = - \frac{1}{2} \left\{ \sum_{i=1}^N \int_a^b \mathcal{U}_{i,K}^2 dx + \sum_{i=1}^N c_{i,K}^2 \int_b^L \omega_i^2(\lambda, x) dx - 1 \right\}. \quad (9a)$$

Таким образом:

$$\begin{aligned} \mu_k = & -\frac{1}{2} \left\{ \sum_{i=1}^N \int_a^b U_{i,k}^2(x) dx + \sum_{i=1}^N C_{i,k}^2 \int_a^b \omega_i^2(\lambda_k x) dx - 1 \right. \\ & + 2 \cdot \sum_{i=1}^N \int_a^b U_{i,k}(x) V_{i,k}(x) dx \left. \right\} \cdot \left\{ \sum_{i=1}^N \int_a^b U_{i,k}(x) W_{i,k}(x) dx \right. \\ & \left. + \sum_{i=1}^N C_{i,k}^2 \omega(\lambda_k x) \frac{\partial}{\partial \lambda} \omega_i(\lambda_k x) dx \right\}^{-1} \end{aligned} \quad (9b)$$

Следовательно, при заданных λ_k и $U_{i,k}(x)$ функция $U_{i,k}(x)$ полностью определена соотношениями (7)-(9). Далее, пользуясь формулами (5), можно вычислить при $t = t_{k+1}$ значения λ_{k+1} и $U_{i,k+1}$. Поскольку начальные значения λ_0 и $U_{i,0}(x)$ при $t = t_0 = 0$ известны, процесс вычисления $\lambda_k, U_{i,k}(x)$ с помощью формул (5)-(9) полностью определен для всех k .

Дискретная схема

Дискретное представление по переменной X рассматриваемой вычислительной схемы может быть получено в результате конечно-разностной аппроксимации краевых задач (8a)-(8b) и применения квадратурных формул того же порядка точности для вычисления интегралов в формуле (9a)-(9b). Для решения краевых задач (8a)-(8b) в основной программе (COSDEQ) вызывается подпрограмма (PROGON), где используются трехточечные конечно-разностные формулы, аппроксимирующие краевую задачу (8a-8b) с точностью порядка $O(h^2)$, где h - шаг по оси X , и

квадратурные формулы трапеций того же порядка точности. Для численного решения дискретных краевых задач применяется алгоритм матричной прогонки^[2]. Матричная прогонка представляет собой обобщение обычного метода прогонки применительно к системе обыкновенных дифференциальных уравнений второго порядка:

$$\vec{U}'' - P(x)\vec{U} + \vec{F}(x) = 0. \quad (10)$$

Здесь \vec{U}, \vec{F} n -мерные векторы, а $P(x)$ - квадратная матрица. Для решения этой системы воспользуемся разностной схемой

$$\frac{\vec{U}_{m+1} - 2\vec{U}_m + \vec{U}_{m-1}}{h^2} - P(x_m)\vec{U}_m + \vec{F}(x_m) = 0, \quad (11)$$

которую перепишем так:

$$\vec{U}_{m+1} - 2 \cdot B_m \vec{U}_m + \vec{U}_{m-1} + \vec{D}_m = 0. \quad (12)$$

В этой записи обозначено:

$$B_m = E + \frac{h^2}{2} P(x_m), \quad (13)$$

E - единичная матрица,

$$\vec{D}_m = h^2 \vec{F}(x_m). \quad (14)$$

Для того чтобы система (12) определяла единственное решение, в нее надо добавить уравнения, отвечающие граничным условиям системы (10), которые мы зададим при $X=0$ и при $X=l$ в следующей форме:

$$\vec{U} \Big|_{X=0} = 0, \quad (15a)$$

$$\frac{d\vec{u}}{dx} + \psi \vec{u} \Big|_{x=e} + \vec{w} = 0, \quad (I5b)$$

где \vec{w} - n -мерный вектор,
 ψ - матрица той же размерности.

Заменив производные, входящие в (I5), разностными отношениями, приходим к разностным граничным условиям, которые могут быть записаны в форме следующих соотношений:

$$\vec{u}_0 = 0, \quad (I6a)$$

$$\vec{u}_{m-1} = G \vec{u}_m + \vec{g}, \quad (I6b)$$

(G - матрица, $M = \frac{e}{h}$).

Разностные уравнения (I2) вместе с граничными условиями (I6) образуют уже полную систему, которая в данной работе решается с помощью подпрограммы PROGN.

Описание программы

Структура программы: главная программа и пакет программ.

Наименование главной программы - **GOJDEQ**,
 подпрограммы - **PROGN**.

Библиотечные подпрограммы: **ABS**, **DETERM**.

Параметр, используемый в данной программе, **PEPS** -
 - характеризует точность вычисления собственного значения и собственной функции задачи (2).

В качестве меры точности пары (λ, \vec{u}) выбрана величина

$$\delta_k = \max_{\substack{2 \leq m \leq M-1 \\ 1 \leq i \leq N}} |h^{-2} (u_{i,k}(x_{m+1}) - 2u_{i,k}(x_m) + u_{i,k}(x_{m-1}))|$$

$$+ |(K_{ij}(x) - \lambda_k) u_{i,k}(x_m) + \sum_{j \neq i}^N K_{ij}(x_m) u_{j,k}(x_m)|.$$

При выполнении условия $\delta_k \leq \text{PEPS}$ происходит выход из программы,

IT - число итерации, или, что то же самое, число шагов по оси x ,

TUMIN - величина минимального шага по оси x . В данной программе для вычисления τ_k применяется формула

$$\tau_k = \begin{cases} \min(1, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \delta_k < \delta_{k-1} \\ \max(\tau_0, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \delta_k \geq \delta_{k-1} \end{cases}$$

В этой формуле τ_0 соответствует параметру TUMIN.

Как показала практика, этот режим обеспечивает наибольшую устойчивость вычислений.

EMAXO и EMAX1 соответствуют величине невязки предыдущей итерации. При первой итерации они полагаются равными некоторой, заранее заданной, константе,

N - число уравнений,

M - число внутренних узловых точек разностной схемы,

AL - начальное приближение к искомому собственному значению,

HL - шаг по оси,

AA, BB - левый и правый концы интервала по оси x .

- в пределах которого мы ищем решения уравнения (2),
- TUO - начальное значение шага по оси t .
 Величина параметра TUO зависит от выбора пробных функций и собственного значения, т.е. от величины невязки при первой итерации.
- NZZ - максимальное число итераций, если при всех итерациях будет иметь место неравенство $\delta_k > PEPS$,
- NZ1 - параметр, соответствующий числу искомых решений, т.е. набору пар искомых собственных значений и собственных функций,
- XIX - параметр, определяющий шаг перехода начального пробного собственного значения к следующему,
- HL - параметр, обусловленный конкретной решаемой задачей, в других случаях его можно положить равным единице,
- NCZ - параметр, определяющий шаг таблицы $U_{i,k}(x)$ при печати промежуточных итераций ($h_{ТАБЛ} = h \cdot NCZ$).

Описание массивов программы COSDEQ

- X(N,M) - массив, соответствующий таблице начального приближения к вычисляемой собственной функции,
- B2(N²,M) - массив, соответствующий таблице коэффициентов

$$K_{ij}(x)$$

- UR(N, M) - массив, соответствующий таблице решений задачи (8) с помощью подпрограммы PROGN,
- AKHN(N) - массив, соответствующий выражению в квадратных скобках (8a-8в),
- XX(N) - соответствует выражению правой части (8a-8в).
 Левое граничное условие в данном случае $V_{i,k}(0) = 0$,
 $W_{i,k}(0) = 0$ задается с помощью столбца AN(N,1)=0.
- F(N,M) - массив, соответствующий правой части задачи (8),
- XVR(N) - массив, соответствующий интегралу от квадрата искомой функции

$$a_i^2 = \int_a^b U_i^2(x) dx,$$
- ALE(IZ1) - массив пробных собственных значений,
- CC(N, IZ1) - массив некоторых заданных коэффициентов, на которые умножаются пробные функции для достижения минимальной невязки при первой итерации. Если они неизвестны, то можно положить их равными единице.

Остальные массивы являются вспомогательными массивами данной программы.

Описание программы COSDEQ и используемых в ней подпрограмм, подпрограмм-функций

В приложении текст программы COSDEQ и подпрограммы PROGON разделены цифрами на линии со звездочками на отдельные блоки. Нижеследующая нумерация будет соответствовать этим цифрам.

Программа COSDEQ

1. В этом блоке вызывается подпрограмма CALD для вычисления таблицы массива $B2(N, M)$, соответствующей матричным элементам

$$K_{ij}(x_m), \quad (m = 1, 2, \dots, M)$$

и BEGIN для вычисления начальных пробных функций. Пробные функции и собственные значения обычно выбираются из физических соображений. При достаточно малых недиагональных матричных элементах

$$|\langle u_i | K_{ii} - \lambda | u_i \rangle| \gg |\langle u_j | K_{ij} | u_i \rangle|$$

можно в качестве пробных функций и собственных значений использовать решения задачи (2) без правой части, т.е. при

$$\sum_{i \neq j} K_{ij}(x) u_j(x) = 0.$$

2. Во втором блоке вычисляется правая часть задачи (8а), т.е. массив F(N, M). Здесь же определяются невязка δ_k (EMAX) и шаг по оси t (TU).

В цикле DO 41 с помощью оператора присвоения

AM(N1, 1)=0, (N1=1, ..., N) определяются левые граничные условия, соответствующие равенству $V_{j,x}(0) = 0, W_{j,x}(0) = 0.$

3. В третьем блоке программы вызываются подпрограммы BOUND1 (AL) (для определения граничного условия (8а) и PROGON (KO) (для решения задачи (8А)). KO - параметр, задающий режим выбора правых граничных условий (8а-8в). ло-

При KO=0, имеют место условия (8а-8в).

При KO≠0, в равенстве (8а-8в) $V_{j,x}'(l) = 0, W_{j,x}'(l) = 0.$

Далее, в цикле DO 8 вычисляется нулевая часть задачи (8в). Подпрограммы BOUND2 (AL) и PROGON (KO) вызываются для вычисления граничного условия (8в) и решения задачи (8) соответственно.

4. Вычисляются интегралы (9а-9в) с помощью квадратурных формул трапеций с точностью порядка $O(h^2)$. Для вычисления интегралов от асимптотических продолжений подынтегральных функций вызывается подпрограмма TAIL (AL).

5. Находятся искомые решения уравнения (2) согласно формулам (5). Вычисляются коэффициенты $a_i^2 = \int u_i^2(x) dx.$

После каждой итерации вычисленная таблица собственных функций и собственное значение выдается на память. Печатаются также число итераций, шаг по оси t, максимальная невязка $\delta_k.$

Приложение I. ПРИМЕР ИСПОЛЬЗОВАНИЯ ПРОГРАММЫ

Настоящая программа была применена для решения конкретной физической задачи - нахождения одночастичных волновых функций и энергий связанного состояния деформированного ядра. При этом необходимо решить уравнение Шредингера

$$\left(-\frac{\hbar^2}{2m} \Delta + V(\vec{r}) - E\right) \psi(\vec{r}) = 0 \quad (\text{II})$$

с эффективным потенциалом взаимодействия

$$V(\vec{r}) = -V_{\text{яд}}(\vec{r}) + V_{s.o.}(\vec{r}) + V_{\text{кул}}(\vec{r});$$

$$V_{\text{яд}}(\vec{r}) = -V_0 \frac{1}{1 + \exp\{d(r - R_0)\}}$$

- ядерное взаимодействие,

$$V_{s.o.}(\vec{r}) = -\kappa [\vec{p} \vec{\sigma}] \text{grad } V_{\text{яд}}$$

- спин-орбитальное взаимодействие,

$$V_{\text{кул}}(\vec{r}) = \frac{3(z-1)e^2}{4\pi R_0^3} \int \frac{n(\vec{r}') d\vec{r}'}{|\vec{r} - \vec{r}'|} \quad (\text{II}2)$$

- кулоновское взаимодействие,

$$n(\vec{r}') = \frac{1}{1 + \exp\{d(r' - R_0)\}}$$

- плотность распределения заряда в ядре.

Проведем в уравнении (II) тождественное преобразование для выделения сферически симметричной части потенциала и в ре-

зультате получим:

$$\left[-\frac{\hbar^2}{2m} \Delta + V(r) + \tilde{V}(\vec{r}) - E\right] \psi(\vec{r}) = 0, \quad (\text{III})$$

$$\text{где } \tilde{V}(\vec{r}) = V(\vec{r}) - V(r). \quad (\text{IV})$$

Далее, предположим, что средний радиус деформированного ядра зависит от параметра деформации $\beta_{\lambda 0}$ и угла относительно оси симметрии

$$R(\theta) = R_0 (1 + \beta_0 + \beta_{20} Y_{20} + \beta_{40} Y_{40}), \quad (\text{V})$$

β_0 - постоянная, введенная для сохранения объема,

β_{20} - квадрупольная деформация,

β_{40} - гексадекапольная деформация,

$R_0 = r_0 A^{1/3} \cdot 10^{-13}$ см - радиус ядра.

Для решения задачи разложим $\tilde{V}(r)$ в ряд по сферическим гармоникам $Y_{\lambda 0}(\theta)$, тогда для $\tilde{V}(\vec{r}) = V(\vec{r}) - V(r)$ будем иметь:

$$\tilde{V}(\vec{r}) = \sum_{\lambda} A_{\lambda}(\beta_{\lambda 0}, r) Y_{\lambda 0}(\theta). \quad (\text{VI})$$

Искомую волновую функцию представим в виде

$$\psi^{\Omega}(\vec{r}) = \sum_{ef} \frac{1}{r} u_{ef}^{\Omega}(r) y_{ef}^{\Omega}, \quad (\text{VII})$$

где

$$y_{ef}^{\Omega} = \sum_{\lambda} (e^{\frac{1}{2}} \Omega - 2 \lambda / \Omega) Y_{e, \Omega-2}^{(\theta, \varphi)} Y_{\lambda 2}.$$

Тогда, так же как и в работе^{/4/}, мы приходим к системе связанных дифференциальных уравнений (2).

Среди всех методов решения уравнения (2) до недавнего времени имел успех метод разложения по определенному полному базису ортонормированных функций. Этот метод обладал несколькими преимуществами: сравнительно малое время счета, удобство передачи информации (с помощью коэффициентов разложения) и т.д. Однако в силу ограниченности числа членов разложения возникала зависимость точности результата от выбора базиса. Очевидно, в этом подходе главную информацию содержали коэффициенты разложения, в частности, главные компоненты разложения. Насколько это так, можно проверить следующим образом.

Запишем условие нормировки в виде

$$\sum_i \int U_i^2(x) dx = 1. \quad (П8)$$

Вклад в сумму интегралов от квадрата каждой функции обозначим через a_i^2 , т.е.

$$a_i^2 = \int U_i(x) dx.$$

Тогда условие (П8) можно переписать в виде

$$\sum a_i^2 = 1. \quad (П8а)$$

Разложим U_i в ряд по какому-либо ортонормированному ба-

зису $\{\varphi_k\}$:
$$U_i = \sum_k a_{i,k} \varphi_k,$$

подставим в (П8), в разложении ограничимся только главными

компонентами, тогда получим:
$$\sum_i (a_{i,k_0})^2 = 1.$$

В табл. I приведены сравнения a_i , вычисленные с помощью данной программы с коэффициентами a_i^0 , взятыми из работы^{/5/}. Из данных таблицы I видно, что главные компоненты разложения в основном уже охватывают искомое решение и могут служить хорошим начальным приближением при решении задачи точным методом.

В заключение авторы выражают благодарность профессору Е.П.Жидкову и сотрудникам его отдела за постоянное внимание к работе и полезные обсуждения.

Авторы также признательны профессору В.Г.Соловьеву и сотрудникам его отдела, С.П.Ивановой и С.А.Гончарову.

ЛИТЕРАТУРА

1. Б.П.Жидков, Г.И.Макаренко, И.В.Пузынин. Непрерывный аналог метода Ньютона в нелинейных задачах физики. ЭЧАЯ 4, 1, 127 (1973).
2. С.К.Годунов, В.С.Рябенский. Введение в теорию разностных схем. Физматгиз. Москва 1962.
3. Л.И.Пономарев, И.В.Пузынин, Т.П.Пузынина. ОИЯИ Р4-6256, Дубна, 1972; ОИЯИ Р4-6919, Дубна 1973; ЖЭТФ, 65, 28 (1973).
4. П.Э.Немировский, В.А.Чепурнов. Ядерная физика, 988, 3, 1966.
5. Ф.А.Гареев, С.П.Иванова, В.Г.Соловьев, С.И.Федотов. Одночастичные энергии и волновые функции потенциала Саксона-Вудса и неротационные состояния нечетных ядер в области $150 \leq A \leq 190$. ЭЧАЯ, 4, 2, 357 (1973).

Рукопись поступила в издательский отдел
10 июля 1974 г.


```

AL=5
AL=AL*HL
AL2=AL
C
C
C SUBROUTINE BOUND1 CALCULATES THE INITIAL FUNCTIONS X(I,J)
C
CALL BOUND1
DO 32 N3=1,N
PRINT 17,(X(N3,IX),IX=1,N,NOZ)
DO 32 I3=1,M11
Y1(N3,I3)=X(N3,I3)
32 CONTINUE
C***** ( 2 ) *****
C
DO 34 IZ=1,I71
DO 35 I3=1,M11
DO 33 N3=1,N
X(N3,I3)=X1(N3,I3)*CC(N3,IZ)
33 CONTINUE
35 CONTINUE
DO 26 NPT=1,127
DO 41 N1=1,N
AN(N1,1)=0
41 CONTINUE
H=H1 * I=1
FK=0
1 CONTINUE
DO 77 NRA=1,N DO 77 NPB=1,N
IJ=(NRA-1)*N+NPB
IF(NRA-NRB) 77,79,77
79 X2(IJ,I)=-32*(NRA,I)-AL
77 CONTINUE
DO 5 H1=1,N
X11(N1,I)=(X(N1,I+2)-2 *X(N1,I+1)+X(N1,I))/(H**2)
6 CONTINUE
DO 2 N1=1,N
A32=0,
DO 3 N2=1,N
N1I2=(N1-1)*I+I2
A32=AB2+B2(N1I2,I)*X(N2,I+1)
1 CONTINUE
F(N1,I)=-A32-X11(N1,I)
2 CONTINUE
DO 28 N1=1,N
AF1=F(N1,I)
AF=ABS(AF1)
IF(AF-FK) 28,23,24
24 BMAX=AF
FK=AF
23 CONTINUE
I=I+1
IF(I-M) 1,16,16
16 I=I+1
TU=TU*BMAX1/BMAX
IF(1.-TU) 24,29,27
29 TU=1.
27 IF(TUMIN-TU) 40,40,42

```

```

42 TUMIN=0.1
40 CONTINUE
C
C***** ( 3 ) *****
C
C SUBROUTINE BOUND1 DETERMINES THE BOUNDARY CONDITIONS
C
CALL BOUND1(AL)
K0=0
CALL PRGON(K0)
R=H1 * I=1
9 DO 8 N1=1,N
F(N1,I)=X(N1,I+1)
X11(N1,I)=JR(N1,I)
8 CONTINUE
R=R+H1
I=I+1
IF(I-M) 9,9,10
10 I=M+1
C
C
C SUBROUTINE BOUND2 DETERMINES THE BOUNDARY CONDITIONS
C
CALL BOUND2(AL)
K0=0
CALL PRGON(K0)
C
C***** ( 4 ) *****
C
ACB=0. * BCA=0.
DO 19 N1=1,N
AKH=3.-2 *H1*AKHH(N1)
V1(N1)=(4.*X11(N1,I-1)-X11(N1,I)+XX1(N1,I)*2.*H1)/AKH
V2(N1)=(4.*UR(N1,I-1)-UR(N1,I)+XX(N1,I)*2.*H1)/AKH
X11(N1,M11)=V1(N1)
UR(N1,M11)=V2(N1)
19 CONTINUE
C
C
C SUBROUTINE TAIL CALCULATES INTEGRALS OF ASYMPTOTICALLY
TAILS OF THE SOLUTIONS OF THIS EQUATIONS
C
CALL TAIL(AL)
DO 15 N1=1,N
ACB=ACB+X(N1,M+1)**2/2 +X(N1,M+1)*V1(N1)
BCA=BCA+X(N1,M+1)*V2(N1)
15 CONTINUE
A3A=OST * BAR=ROS
XM(M+1)=ACB
XXM(M+1)=BCA
DO 13 I=1,M
A9=0. * 3A=0
DO 14 N1=1,N
A9=AB+X(N1,I)*X11(N1,I)+X(N1,I)**2/2.
BA=BA+X(N1,I)*UR(N1,I)
14 CONTINUE
XM(I)=A9
XXM(I)=BA
13 CONTINUE
H=H1

```

```

DO 20 I=1,M
ABA=ABA+(X(I)+X(I+1))*H/2
BAB=BAB+(XX(I)+XX(I+1))*H/2.
20 CONTINUE
ABA=ABA-0.5
AMJ=-ABA/BAB
C
C***** ( 5 )*****
C
DO 12 I=1,111
DO 12 N1=1,N
X(N1,I)=(J2(N1,I)*AMU+X11(N1,I))*TU+X(N1,I)
12 CONTINUE
AL=AL+TU*A4U
AL1=-AL/HL
PRINT 50,IT,BMAX,TU,AL1
IT=IT+1
DO 38 N1=1,N
PRINT 17,(X(N1,IX),IX=1,M,NCZ)
34 CONTINUE
DO 36 N1=1,N
AKR=0.
DO 37 I1=1,M11
AKR=AKR+X(N1,I1)**2*H1
37 CONTINUE
XVR(N1)=AKR
36 CONTINUE
PRINT 17,(XVR(IX),IX=1,N)
BMAX1=BMAX
IF (BMAX-P:PS) 140,140,26
26 CONTINUE
140 CONTINUE
AL2=AL2+XIX*HL
AL=AL2
AL=ALF(I2)*HL
BMAX1=BMAX1
TU=TU0
34 CONTINUE
17 FORMAT (2X,5E20.10)
50 FORMAT (3X5HTTFR=T4,3X6HDELTA=E16.9,5X4HTAU=E16.9,
15X15HE(E15:N VALUE)=E15.9)
END

```

```

SUBROUTINE PROGON(K0)
C
C*****
C
SUBROUTINE PROGON FINDS OF THE SOLUTIONS OF THE COUPLED
SYSTEM OF DIFFERENTIAL EQUATIONS BY THE METHOD OF ALTERNATING
DIRECTION IMPLICIT
C
C*****
C
COMMON/HX/ M,AA,BB,H1
COMMON/K/ MA,N,NN
COMMON/UCR/ UR(6,120),F(6,120),AN(6,120)
COMMON/AXX/ XX(6),AKHH(6)
COMMON/B2B2/ B2(36,120)
DIMENSION AKH1(6),AKH(5),B3(36),R1(36),A1(36),A2(36),AKH2(6),UC(5)
DIMENSION A(36,120)
C
H=H1
NK=1
NKK=NK
M21=M-1
N1=N-1
NN1=N1*N1
DO 63 I=1,N
DO 63 J=1,N
IJ=(I-1)*N+J
A(IJ,1)=0.
63 CONTINUE
C
C***** ( 1 )*****
C
11 DO 1 I=1,N
DO 1 J=1,N
IJ=(I-1)*N+J
JI=(J-1)*N+I
IF(I-J) 3,2,3
2 B1(IJ)=2.-B2(IJ,NK)*H1*H1-A(JI,NKK)
GO TO 1
3 B1(IJ)=-B2(IJ,NK)*H1*H1-A(JI,NKK)
1 CONTINUE
NKK=NK+1
DO 100 I=1,N
F(I,NK)=F(I,NK)*H1*H1
100 CONTINUE
DO 21 I=1,N
DO 21 J=1,N
IJ=(I-1)*N+J
B3(IJ)=B1(IJ)
21 CONTINUE
IF(N-2) 103,103,104
104 BK=DETERM(B1,N,N)
GO TO 105
103 CONTINUE
BK=B1(1)*B1(4)-B1(3)*B1(2)
105 DO 5 I=1,N
DO 34 I1=1,N

```

```

33 DO 34 J1=1,N
    IJ1=(I1-1)*N+J1
    IF (J1-1) 35,35,35
36 A2(IJ1)=A1(I1,NK)-F(I1,NK)
    GO TO 34
35 A2(IJ1)=B3(IJ1)
34 CONTINUE
    IF (N-2) 105,105,107
107 AK=DETERM(A2,N,N)
    GO TO 158
195 CONTINUE
    AK=A2(1)*A2(4)-A2(3)*A2(2)
198 CONTINUE
    AN(I,NKK)=AK/3K
    DO 5 J=1,N
        IJ=(I-1)*N+J
        IJ1=1 9 IJ2=1
        DO 5 I1=1,I
            DO 5 J1=1,N
                IJ2=IJ2+1
                IF (I-I1) 8,8,8
            9 IF (J-J1) 7,6,7
            7 A1(IJ1)=B3(IJ2)
            AK=B3(IJ2)
            IJ1=IJ1+1
        6 CONTINUE
        IF (N1-1) 101,101,102
102 AK=DETERM(A1,I1,N1)
101 A(IJ,NKK)=(I-1)*(I+J)-AK/BK
    5 CONTINUE
    H=H+H1
    NK=NK+1
    IF (NK-M) 11,17,17
C
C***** ( 2 )*****
C
17 DO 99 I=1,N
    AKH2(I)=XX(I)*?.*H1/(AKHH(I)*H1*2.-3.)
    AKH(I)=4./ (AKHH(I)*H1*2.-3.)
    AKH1(I)=1./ (AKHH(I)*H1*2.-3.)+1
90 CONTINUE
    IF (K0) 57,56,57
56 DO 58 I=1,N
    DO 58 J=1,N
        IJ=(I-1)*N+J
        JI=(J-1)*N+I
        IF (I-J) 1+1,42,1+1
42 B1(IJ)=B3(IJ)+AKH(1)+A(IJ,NK-1)*(1.-AKH1(I))
    GO TO 58
141 B1(IJ)=B3(IJ)+4*(JI,NK-1)*(1.-AKH1(I))
58 CONTINUE
    DO 70 I=1,N
    DO 71 I1=1,N
    DO 71 J1=1,N
        IJ1=(I1-1)*N+J1
        IF (J1-1) 73,74,73
74 A2(IJ1)=AN(I1,NK-1)*AKH1(I1)-F(I1,NK-1)
        A2(IJ1)=A2(IJ1)-AKH2(I1)

```

```

33 DO 71
34 A2(IJ1)=B1(IJ1)
31 CONTINUE
    IF (N-2) 104,103,111
110 AN(I,NK)=DETERM(A2,N,N)
    GO TO 70
109 AN(I,NK)=A2(1)*A2(4)-A2(3)*A2(2)
70 CONTINUE
    IF (N-2) 112,112,113
113 BKK=DETERM(B1,N,N)
    GO TO 114
112 CONTINUE
    BKK=B1(1)*B1(4)-B1(3)*B1(2)
114 DO 60 I=1,N
    AN(I,NK)=A1(I,NK)/BKK
    60 CONTINUE
C
C***** ( 3 )*****
C
57 H=H+H1
18 NKK=NK-1
    DO 13 I=1,N
        UK=0.
        IF (KU) 53,54,53
53 DO 14 J=1,N
        IJ=(I-1)*N+J
        JI=(J-1)*N+I
        UK=JK+A(IJ,NK)*UC(J)
14 CONTINUE
        UR(I,NK)=JK+AN(I,NK)
        GO TO 13
54 UR(I,NK)=AN(I,NK)
13 CONTINUE
    K0=2
    IF (NKK) 15,16,19
19 DO 15 I=1,N
    UC(I)=UR(I,NK)
15 CONTINUE
    H=H+H1
    NK=NK-1
    IF (NK-1) 15,16,15
16 CONTINUE
    RETURN
END

```