

# **SYSINT (SYSINTM) – комплекс программ для решения задачи на собственные значения для системы интегральных уравнений**

Программный комплекс SYSINT (SYSINTM) предназначен для решения задачи на собственные значения для системы интегральных уравнений. Реализована модифицированная итерационная схема, реализована на основе непрерывного аналога метода Ньютона (НАМН). Приводится описание алгоритмов и параметров программ комплекса, рассматривается численный пример. Более подробное описание программного пакета дано в [1].

## **SYSINT(SYSINTM) – program complex for numerical solution of the eigenvalue problem for the system of integral equations**

Program complex SYSINT(SYSINTM) was developed for the numerical solution of the eigenvalue problem for the system of integral equations. The iteration scheme is constructed on the basis of the modified continuous analog of the Newton method. Algorithms, parameters of the program, and examples are presented. Detailed description is given in [1].

### **1 Назначение и общая характеристика пакета**

Программный комплекс SYSINT(SYSINTM) [1] предназначен для численного решения системы  $L$  интегральных уравнений следующего вида:

$$\vec{\Phi}(z) = \hat{Q}(x)\vec{\phi}(x) - \lambda\hat{R}(x)\vec{\phi}(x) + \int_0^R \hat{K}(x, x')\vec{\phi}(x')dx' = 0, \quad (1)$$

$$z = (\lambda, \vec{\phi}(x)), \quad \vec{\phi} = \{\phi_1(x), \phi_2(x), \dots, \phi_L(x)\}$$

с условием нормировки

$$\Gamma(z) = \int_0^R dx \sum_{l=1}^L \phi_l^2(x) - G = 0, \quad (2)$$

где

$$\hat{Q}(x) = \begin{pmatrix} Q_{11}(x) & \dots & Q_{1L}(x) \\ \dots & \dots & \dots \\ Q_{L1}(x) & \dots & Q_{LL}(x) \end{pmatrix}, \quad \hat{R}(x) = \begin{pmatrix} R_{11}(x) & \dots & R_{1L}(x) \\ \dots & \dots & \dots \\ R_{L1}(x) & \dots & R_{LL}(x) \end{pmatrix},$$

$$\hat{K}(x, x') = \begin{pmatrix} K_{11}(x, x') & \dots & K_{1L}(x, x') \\ \dots & \dots & \dots \\ K_{L1}(x, x') & \dots & K_{LL}(x, x') \end{pmatrix},$$

$Q_{lm}(x)$ ,  $R_{lm}(x)$ ,  $K_{lm}(x, x')$  – заданные функции, обеспечивающие существование нетривиальных решений  $z^* = (\lambda^*, \vec{\phi}^*(x))$ .

Итерационные схемы разработаны на основе обобщенного непрерывного аналога метода Ньютона [2]. В программе SYSINTM реализована модификация с одновременным обращением оператора производной Фреше, предложенная в [3] и обеспечивающая существенный выигрыш во времени при расчетах на векторных вычислительных системах [3, 1].

Программы были проверены на ряде тестовых задач. Комплекс широко использовался для численного исследования системы Бете–Солпитера и релятивистского обобщения уравнения Шредингера в моделях кваркония (см. обзор [2] и цитируемую литературу).

Ниже дается описание алгоритмов и параметров комплекса, рассматриваются особенности программной реализации и приводятся численные примеры, иллюстрирующие его работу.

## 2 Алгоритм программы SYSINT

Согласно подходу, определяемому обобщенным НАМН, нелинейное функциональное уравнение в  $B$  – пространстве

$$\varphi(z) \equiv \begin{pmatrix} \vec{\Phi}(z) \\ \Gamma(z) \end{pmatrix} = 0, \quad (3)$$

представляющее исследуемую задачу, заменяется эволюционным уравнением по непрерывному параметру  $t$

$$\frac{d}{dt}\varphi(t, z(t)) = -\varphi(t, z(t)), \quad 0 \leq t < \infty$$

с начальным условием  $z(0) = z_0$ . Обозначив  $\hat{A}(t) = \vec{\Phi}'_{\vec{\phi}}(t, z(t))$ ,  $z(t) = (\lambda(t), \vec{\phi}(x))$ , получим для уравнения (1):

$$\hat{A}(t) \frac{d\vec{\phi}(x, t)}{dt} = \lambda'_t(t) \hat{R}(x) \vec{\phi}(x, t) - \vec{\Phi}(t, z(t)). \quad (4)$$

Вводя дискретную сетку по непрерывному параметру  $t : \{0 = t_0 < t_1 < \dots < t_k < \dots\}$  и обозначая  $z_k = z(t_k)$ ,  $\lambda_k = \lambda(t_k)$ ,  $\vec{\phi}_k(x) = \vec{\phi}(x, t_k)$ ,  $\hat{A}_k = \hat{A}(t_k)$ ,  $\vec{\Phi}_k = \vec{\Phi}(t_k, z(t_k))$ ,  $\vec{v}_k(x) = \vec{\phi}'_t(x, t_k)$ ,  $\mu_k = \lambda'_t(t_k)$ , после аппроксимации задачи (4) по схеме Эйлера получаем формулы, определяющие итерационный процесс:

$$\begin{aligned} \vec{\phi}_{k+1}(x) &= \vec{\phi}_k(x) + \tau_k \vec{v}_k(x), \\ \lambda_{k+1} &= \lambda_k + \tau_k \mu_k, \end{aligned} \quad (5)$$

где

$$\vec{v}_k(x) = -\vec{v}_k^{(1)}(x) + \mu_k \vec{v}_k^{(2)}(x), \quad (6)$$

$$\hat{B}_k = \hat{A}_k^{-1}, \quad (7)$$

$$\vec{v}_k^{(1)}(x) = \hat{B}_k \vec{\Phi}_k, \quad (8)$$

$$\vec{v}_k^{(2)}(x) = \hat{B}_k \hat{R}(x) \vec{\phi}_k(x), \quad (9)$$

$\mu_k$  вычисляется из условия нормировки (2):

$$\mu_k = \frac{G - \int_0^R dx \sum_{l=1}^L (\phi_l)_k^2(x) + 2 \int_0^R dx \sum_{l=1}^L (v_l^{(1)})_k(x) (\phi_l)_k(x)}{2 \int_0^R dx \sum_{l=1}^L (v_l^{(2)})_k(x) (\phi_l)_k(x)}. \quad (10)$$

Вычисляя для каждого значения  $t_k$  итерационные поправки  $\vec{v}_k(x)$ ,  $\mu_k$  и шаг  $\tau_k$  (см. разд.5), получаем новое приближение  $z_{k+1}$  к решению  $z^*$ .

Итерационный процесс должен продолжаться до тех пор, пока не будет выполнено соотношение

$$\delta_k = \|\vec{\Phi}_k\| \leq \epsilon, \quad (11)$$

где  $\epsilon > 0$  – заранее заданное малое число, а невязка  $\delta_k$  вычисляется по формуле:

$$\delta_k = \max_{l=1,L} \max_x |(\Phi_l)_k| \quad (12)$$

### 3 Алгоритм программы SYSINTM

Перейдем к описанию модифицированного алгоритма. Вместо уравнения (3) рассматривается система следующего вида:

$$\begin{aligned} \varphi(z) &= 0, \\ \mathcal{B}\mathcal{A} - \mathcal{I} &= 0, \end{aligned} \quad (13)$$

где  $\mathcal{A} = \varphi'_z(z)$ ,  $\mathcal{B} = \mathcal{A}^{-1}$ ,  $\mathcal{I}$  – единичный оператор.

Вводя непрерывный параметр  $t (0 \leq t < \infty)$  и переходя к системе эволюционных уравнений, получаем:

$$\begin{aligned} \frac{d}{dt} \varphi(t, z(t)) &= -\varphi(t, z(t)), \\ \frac{d}{dt} [\mathcal{B}(t)\mathcal{A}(t) - \mathcal{I}] &= \mathcal{I} - \mathcal{B}(t)\mathcal{A}(t). \end{aligned} \quad (14)$$

В соответствии с (14) для уравнения (1) имеем:

$$\begin{aligned} \hat{A} \frac{d\vec{\phi}(x, t)}{dt} &= -\lambda'_t(t) \hat{R}(x) \vec{\phi}(x, t) - \vec{\Phi}(t, z(t)), \\ \frac{d}{dt} [\hat{B}(t)\hat{A}(t) - \hat{I}] &= \hat{I} - \hat{B}(t)\hat{A}(t). \end{aligned} \quad (15)$$

После дискретизации непрерывного параметра  $t$  получаем на основе схемы Эйлера следующую итерационную схему:

$$\begin{aligned}\vec{\phi}_{k+1}(x) &= \vec{\phi}_k(x) + \tau_k \vec{v}_k(x), \\ \lambda_{k+1} &= \lambda_k + \tau_k \mu_k, \\ \hat{B}_{k+1} &= \hat{B}_k + \tau_k \hat{W}_k,\end{aligned}\tag{16}$$

где  $\vec{v}_k(x)$  вычисляется с помощью формул (6),(8),(9), а  $\mu_k$  определяется по формуле (10).

Формула для итерационной поправки  $\hat{W}_k$  имеет вид:

$$\hat{W}_k = [\hat{I} - \hat{B}_k(\hat{A}_k + \hat{A}'_{kt})]\hat{B}_k.\tag{17}$$

$\hat{W}_k$  может также вычисляться по более простой формуле, полученной на основе работы [4]:

$$\hat{W}_k = [\hat{I} - \hat{B}_k \hat{A}_k] \hat{B}_k.\tag{18}$$

Однако при этом, как показали вычисления, сужается область сходимости итерационной схемы.

Таким образом, имея начальное приближение  $z_0$  и  $\hat{B}_0$ , можно последовательно найти все приближения  $z$  и  $\hat{B}_k$ . Итерационный процесс продолжается до выполнения неравенства (11).

В качестве начального приближения  $\hat{B}_0$  используется  $\hat{B}_0 = \hat{A}^{-1}(z_0)$ .

## 4 Программная реализация

Программы написаны на языке FORTRAN.

Для вычисления  $\hat{B}_k = \hat{A}_k^{-1}$  в настоящей версии используются стандартные подпрограммы из библиотеки LINPACK.

Аппроксимация системы (1) на дискретной сетке по аргументу  $x$ , ( $0 = x_1, x_2, \dots, x_N = R$ ) с шагом  $h_i = x_{i+1} - x_i$  имеет в точке  $x_i$  ( $i = 1, \dots, N$ ) следующий вид:

$$\begin{aligned}\Phi_l(\lambda, \vec{\phi}(x_i)) &= \sum_{m=1}^L (Q_{lm}(x_i) - \lambda R_{lm}(x_i)) \phi_m(x_i) + \\ &+ \sum_{m=1}^L \sum_{j=1}^N K_{lm}(x_i, x_j) \phi_m(x_j) \xi_j = 0, \quad l = 1, \dots, L.\end{aligned}\tag{19}$$

Коэффициенты  $\xi_j$  зависят от способа численного интегрирования. При этом порядок сходимости численного решения  $z_k$  к точному решению  $z^*$  соответствует порядку точности выбранной квадратурной формулы, что подтверждается расчетами на последовательности сгущающихся сеток [3, 1].

## 5 Алгоритмы вычисления параметра $\tau_k$ .

Для вычисления итерационного параметра  $\tau_k$  в программе реализованы следующие алгоритмы .

$$1. \quad \tau_k \equiv \tau_0; \quad 0 < \tau_0 \leq 1. \quad (20)$$

$$2. \quad \tau_k = \begin{cases} \min(1, 2\tau_{k-1}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1}/2), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (21)$$

Величина  $\delta_k$  везде вычисляется по формуле (12)

$$3. \quad \tau_k = \begin{cases} \min(1, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k \geq \delta_{k-1} \end{cases} \quad (22)$$

$$4. \quad \tau_k = \frac{\delta_{k-1}^2}{\delta_{k-1}^2 + \delta_k(1)^2}, \quad (23)$$

где  $\delta_k(1)$  – невязка на  $k$ -той итерации для  $\tau_k = 1$ .

5. На равномерной сетке  $\omega_\tau$  отрезка  $[0, 1]$  с шагом  $\Delta\tau$  вычисляется последовательность невязок  $\delta^i$  по формуле (12) и выбирается такое значение  $\tau_k$ , которому соответствует минимальная невязка.

## 6 Подпрограммы пользователя

Подпрограмма SUBROUTINE KSI(N,X,TKSI), предназначенная для вычисления коэффициентов  $\xi_j$ , должна быть составлена пользователем. Здесь N и X – соответственно число точек и массив узлов дискретной сетки по аргументу  $x$ , TKSI – массив коэффициентов  $\xi_j$ . Массивы X и TKSI имеют размерность N.

Пользователь должен также составить и включить в комплекс следующие подпрограммы - функции:

QQ(L,M,XI) для вычисления  $Q_{lm}(x_i)$ ,

RR(L,M,XI) для вычисления  $R_{lm}(x_i)$ ,

YK(L,M,XI,XJ) для вычисления  $K_{lm}(x_i, x_j)$ .

## 7 Описание параметров программы

Обращение к комплексам SYSINT и SYSINTM осуществляется операторами  
CALL SYSINT (N, LM, LMN, X, EV, FI, TAU0, NTAU, HT5, GNORM, EPS, NMAX, F, V, V1, R1, R2, R3, TKSI, B),

CALL SYSINTM (N, LM, LMN, X, EV, FI, TAU0, NTAU, GNORM, EPS, NMAX, F, V, V1, R1, R2, R3, TKSI, B, A, DAT, NMOD).

N – число узлов сетки по аргументу  $x$ .

LM – число уравнений системы (1).

LMN – размерность рабочих массивов, LMN=N·LM.

X – массив узлов сетки по  $x$  размерности N.

EV – собственное значение  $\lambda$ . При обращении к программе этому параметру присваивается начальное приближение для собственного значения  $\lambda_0$ , после окончания работы программы здесь находится  $k$ -е приближение  $\lambda_k$ .

FI – двумерный массив размерности LM×N. При обращении к программе в нем задается начальное приближение  $(\phi_l)_0(x_i)$ , ( $l = 1, \dots, L$ ,  $i = 1, \dots, N$ ), на выходе здесь находится полученное решение  $(\phi_l)_k(x_i)$ .

TAU0 – начальное значение шага  $\tau_0$ .

NTAU – номер алгоритма вычисления  $\tau_k$  (разд. 5)

При NTAU=1 используется формула (20)

При NTAU=2 используется формула (21)

При NTAU=3 используется формула (22)

При NTAU=4 используется формула (23)

При NTAU=5 используется алгоритм 5 разд.5.

NT5 – шаг сетки  $\omega_\tau$  из алгоритма 5. Для NTAU≠5 этот параметр не используется.

GNORM – значение  $G$  из условия нормировки (2).

EPS – заданное малое число  $\epsilon > 0$  из соотношения (11).

NMAX – максимально допустимое число итераций  $k$ . При его превышении происходит выход из программы.

F, V, V1, R1, R2, R3, B, TKSI – рабочие массивы. Массивы F, V, V1, R1, R2, R3 имеют размерность LMN. Массив TKSI имеет размерность N. Двумерный массив B имеет размерность LMN×LMN.

A, DAT – дополнительные двумерные рабочие массивы программы SYSINTM, также имеющие размерность LMN×LMN.

NMOD – параметр модифицированного алгоритма. При NMOD=0 итерационная поправка  $\hat{W}_k$  вычисляется по формуле (17). Если NMOD≠0, то  $\hat{W}_k$  вычисляется по формуле (18).

## 8 Численный пример

Тестовая задача для системы (1) при  $L = 2$  имеет следующий вид :

$$\hat{Q}(x) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \hat{R}(x) = \begin{pmatrix} 1 & 0 \\ 0 & 1 + x - \frac{1}{x} \end{pmatrix}, \hat{K}(x, x') = \begin{pmatrix} x' & -1 \\ \frac{x^3}{x'} & \frac{-x}{x'^2} \end{pmatrix}. \quad (24)$$

Условие нормировки  $\int_0^R dx(\phi_1^2(x) + \phi_2^2(x)) - G = 0$ ,  $G = 1/3 + 1/5$ ,  $R = 1$ .

Аналитическое решение задачи (24)  $\lambda^* = 1$ ,  $\phi_1^*(x) = x$ ,  $\phi_2^*(x) = x^2$ . Вычисления проводились на равномерной сетке по  $x$  с шагом  $h = 0.05$ . Коэффициенты  $\{\xi_j\}$

вычислялись в соответствии с формулой Симпсона. Результаты счета при начальном приближении  $\lambda_0 = .9$ ,  $\phi_{10}(x) = x^1.1$ ,  $\phi_{20}(x) = x^2.1$ , NTAU=3, TAU0=0.1, EPS=0.00001 приведены в таблице 1. Начальное значение невязки  $\delta_0 \simeq 0.103$ .

ТАБЛИЦА 1

PROGRAM	k	$\lambda_k$	$\delta_k$
SYSINT	11	0.999999987	$3.8 \cdot 10^{-6}$
SYSINTM	13	0.999999983	$3.3 \cdot 10^{-6}$

Сравнительные расчеты на векторной ЭВМ CONVEX C120 [3, 1] показывают, что по временным характеристикам модифицированный алгоритм, реализованный в программе SYSINTM, является более эффективным для векторной вычислительной системы. При расчетах на последовательных вычислительных системах более более быстройдействующей является программа SYSINT.

## Список литературы

- [1] Земляная Е.В. SYSINT(SYSINTM) – комплекс программ для численного решения задачи на собственные значения для системы интегральных уравнений. Сообщение ОИЯИ P11-94-120, Дубна, 1994.
- [2] Пузынин И.В., Амирханов И.В., Земляная Е.В., Первушин В.Н., Пузынина Т.П., Стриж Т.А., Лахно В.Д. Обобщенный непрерывный аналог метода Ньютона для численного исследования некоторых квантово-полевых моделей. ЭЧАЯ, 1999, т.30, вып.1, стр.210-265.
- [3] Puzynin I.V., Amirkhanov I.V., Puzynina T.P., Zemlyanaya E.V. The Newtonian Iterative Scheme with Simultaneous Calculating the Inverse Operator for the Derivative of Nonlinear Function. JINR Rapid Comm., No.5[62]-93, Dubna, 1993, p.63-73; In: Proc. Intern. Conf. Programming and Mathematical Methods for Solving Physical Problems. P&MM'93, Ed. by Yu.Yu.Lobanov and E.P.Zhidkov, Dubna, 1993, p.30-34.
- [4] Давиденко Д.Ф. Препринт ИАЭ им. И.В.Курчатова, ИАЭ-1963, Москва, 1970.