287-00

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

Р11-2000-287

Г.А.Емельяненко, Э.Б.Душанов[1], М.Г.Емельяненко[2],
Т.Т.Рахмонов[1], А.П.Сапожников

МАШИННО-НЕЗАВИСИМЫЙ ПАКЕТ ПРОГРАММ JINRLINPACK ДЛЯ РЕШЕНИЯ ПЛОХО ОБУСЛОВЛЕННЫХ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

[1]Институт ядерной физики АН Республики Узбекистан, Ташкент
[2]Кафедра вычислительной математики и кибернетики МГУ

2000

Введение

В настоящей работе приводится описание нового машинно–независимого пакета программ JINRLINPACK на Фортране–77, в котором реализованы алгоритмы метода критических компонент [1–4] решения плохо обусловленных систем линейных алгебраических уравнений и получения констант вещественной арифметики ЭВМ [5] и который поставлен [6] в библиотеку LIBJINR базовых ЭВМ ОИЯИ. Фортранный текст всех программ пакета JINRLINPACK находится в файле f499.f на сервере Convex и доступен через WWW по адресу http:/www.jinr.ru/~tsap/Koi/jinrlib. В пакет входят программы:

INIT — получения констант вещественной арифметики ЭВМ;
Lin2dsysccmSolver — решения систем линейных уравнений $C_2X = Y$;
Lin3dsysccmSolver — решения систем линейных уравнений $C_3X = Y$;
LinsysccmSolver,PseudsysccmSolver — решения систем линейных алгебраических уравнений $AX = Y$.

В работе приводятся также тексты программ пакета на Фортране–77, наиболее характерные тестовые примеры плохо обусловленных систем и результаты тестовых расчетов. Сравнение результатов численных экспериментов, выполненных с использованием нового пакета, с результатами наиболее известных программ из пакетов:

CERNLIB — библиотека программ CERN [7];
NAG — пакет математических программ (Numerical Algorithms Group, Oxford) [8];
LIBJINR — библиотека программ ОИЯИ [9];
LINA — пакет программ [10],

подробно представлено в [2–4].
Приводятся примеры тестирования модулей пакета JINRLINPACK.

1. Описание программы INIT — получения констант вещественной арифметики ЭВМ. Примеры тестовых расчетов

В этом параграфе приводится описание программ:

INIT — получения констант вещественной арифметики ЭВМ, а также получения параметров настройки функций GTEXP,CHEXP;
GTEXP — выделения мантиссы и порядка вещественного числа;
CHEXP — восстанавления вещественного числа по его мантиссе и порядку.

Параметры настройки функций GTEXP,CHEXP получаются в INIT в общем блоке ICHGT, в котором хранится полная информация о формате представления нормализованного числа в ЭВМ. При этом, если содержимое блока ICHGT уже известно пользователю, то он может самостоятельно сформировать этот блок (см. структуру блока ICHGT ниже в описании программ INIT) и обращение к функциям GTEXP,CHEXP может быть осуществлено без предварительной работы программы INIT. В противном случае перед обращением к функциям GTEXP,CHEXP сначала вызывается INIT. В настоящем параграфе приводятся также

– примеры использования INIT,GTEXP и CHEXP;
– некоторые тестовые результаты работы программ INIT,GTEXP и CHEXP.

Описание программ INIT, GTEXP и CHEXP

Программа INIT

Структура: SUBROUTINE

Внешние подпрограммы и функции: INITP – подпрограмма печати вычисленных машинных констант.

Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV

Обращение: CALL INIT, если печать вычисленных констант не требуется;
CALL INITP, если константы печатаются.

Входные данные: нет (т. к. подпрограмма определяет базовые константы данной ЭВМ).

Выходные данные: вещественные константы $\varepsilon_\infty, \varepsilon_0, \varepsilon_1, \varepsilon_2 = \varepsilon_1/\beta$ и целые константы β, u, l, t. Выдаются в блоках RCONST и ICONST в указанных последовательностях соответственно. В блоке ICHGT содержатся следующие выделенные параметры настройки функций GTEXP, CHEXP:

- BINV – (real*8) вещественное число, равное $1/\beta$;
- KB – (integer) номер половины формата, в котором хранится младший разряд порядка нормализованного числа;
- LWT – (integer) представление целого порядка $e = 0$ в разрядах формата, отведенных для порядка;
- INO – (integer) количество β-ичных разрядов, отведенных для порядка с учетом его знака (в формате для хранения вещественного числа);
- INB – (integer) количество β-ичных разрядов, отведенных для мантиссы с учетом ее знака (в формате для хранения вещественного числа);
- MNE1 – (integer) признак кода ($e > 0$)–положительного порядка. При этом MNE1 $= [e]^k_{код} - e$;
- MNE2 – (integer) признак кода ($e < 0$)–отрицательного порядка. При этом MNE2 $= [e]^k_{код} - e$.

Метод: алгоритм автоматического получения констант вещественной арифметики ЭВМ, изложенный в [5].

Печать: подпрограмма INITP вычисляет и печатает константы вещественной арифметики ЭВМ.

Программа GTEXP

Структура: DOUBLE PRECISION FUNCTION

Внешние подпрограммы и функции: нет.

Общие блоки: COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV

Обращение: RM=GTEXP(R,E)

Входные данные: R – (real*8) вещественное число, порядок и мантисса которого выделяется; блок ICHGT содержит настроечные параметры, определенные в INIT (или заранее известные).

Выходные данные: RM – (real*8) содержит выделенную мантиссу числа R; E – (integer) содержит выделенный порядок числа R.

Метод: логические операции по выделению мантиссы и порядка с учетом информации блока ICHGT.

Программа CHEXP

Структура: DOUBLE PRECISION FUNCTION

Внешние подпрограммы и функции: нет.

Общие блоки: COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV

Обращение: R=CHEXP(RM,E)

Входные данные: RM – (real*8) мантисса числа R; E – (integer) порядок числа R; блоки ICONST и ICHGT содержат элементы, определенные в INIT (или заранее известные).

Выходные данные: R – (real*8) содержит вещественное число $\beta^{\mathrm{E}}\cdot$ RM=R.

Метод: логические операции по восстановлению вещественного числа по мантиссе и порядку с учетом информации блоков ICONST, ICHGT.

Пример *(тест)* **1.** Пусть, например, ЭВМ есть IBM PC. Известно, что машинные константы компьютеров такого типа есть: $\beta = 2, t = 53, l = -1021, u = 1024$. Построим программу TEST, которая, используя программы INIT, GTEXP и CHEXP, восстановит эти константы, а также константы $\varepsilon_1 = \frac{1}{\beta}\cdot\beta^{2-t}$ — относительная погрешность машинной арифметики, $\varepsilon_0 = \frac{1}{\beta}\cdot\beta^l$ — "машинный" нуль и $\varepsilon_\infty = (1 - \frac{1}{\beta^t})\cdot\beta^u$ — "машинная" бесконечность и при этом выделит мантиссы и порядки вещественных констант: $1, \varepsilon_1, \varepsilon_0, \varepsilon_\infty$, а также обратно восстановит эти константы по выделенным мантиссам и порядкам.

Ниже приводится текст программы этой задачи, а также таблицы 1, 2 и 3 — результаты работы программы. При этом в программе TEST использована подпрограмма SUB, которая восстанавливает одновременно группу указанных чисел по выделенной мантиссе и порядку. Программа TEST печатает результаты своей работы — таблицы 1, 2, 3.

```
      PROGRAM TEST
      IMPLICIT REAL*8 (A-H, O-Z)
      INTEGER RADIX
      COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
      COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
      CALL INITP
      D1=GTEXP(1D0,I)
      DE=GTEXP(EPS,J)
      DO=GTEXP(OVFLO,L)
      DU=GTEXP(UNDFLO,K)
      D=1D0
      PRINT1,D,D1,I,EPS,DE,J,UNDFLO,DU,K,OVFLO,DO,L,
    1 FORMAT(10X,'ЧИСЛО',10X,':',6X,'МАНТИССА',6X,': ПОРЯДОК',/,
     * 25('-'),
     * '+',20('-'),'+',8('-'),4(/,E24.15E3,' :',F19.16,' : ',I5),/)
      CALL SUB(D1,I,DE,J,DO,L,DU,K)
```

```
      STOP
      END
C
      SUBROUTINE SUB(D,I,DE,J,DO,L,DU,K)
      IMPLICIT REAL*8 (A-H, O-Z)
      D1=CHEXP(D,I)
      EPS=CHEXP(DE,J)
      OVFL=CHEXP(DO,L)
      UNDF=CHEXP(DU,K)
      PRINT1,D,I,D1,DE,J,EPS,DE,K,UNDF,DO,L,OVFL
    1 FORMAT(6X,'МАНТИССА',6X,': ПОРЯДОК:',10X,'ЧИСЛО',/,20('-'),
     * '+',8('-'),'+',25('-'),4(/,F19.16,' : ',I5,'  :',E24.15E3))
      RETURN
      END
```

Таблица 1 *констант вещественной арифметики ЭВМ:*

```
The constants of mashin's arithmetic ...

MAXEXP=    1024   OVFLO=  1.797693134862316E+308
MINEXP=   -1021  UNDFLO=  2.225073858507201E-308
 RADIX=       2     EPS=  2.220446049250313E-016
MANTSZ=      53  EPSMIN=  1.110223024625157E-016
```

Таблица 2 *выделенных мантисс и порядков вещественных чисел:* $1 = \frac{1}{\beta}\cdot\beta^1$, $\varepsilon_1 = \frac{1}{\beta}\cdot\beta^{2-t}, \varepsilon_0 = \frac{1}{\beta}\cdot\beta^l$ и $\varepsilon_\infty = (1-\frac{1}{\beta^t})\cdot\beta^u$

```
        ЧИСЛО            :      МАНТИССА       : ПОРЯДОК
-------------------------+---------------------+--------
  .100000000000000E+001 :  .5000000000000000 :      1
  .222044604925031E-015 :  .5000000000000000 :    -51
  .222507385850720E-307 :  .5000000000000000 :  -1021
  .179769313486232E+309 :  .9999999999999999 :   1024
```

Таблица 3 *восстановленных по мантиссам и порядкам вещественных чисел:* $\frac{1}{\beta}\cdot\beta^1 = 1, \frac{1}{\beta}\cdot\beta^{2-t} = \varepsilon_1, \frac{1}{\beta}\cdot\beta^l = \varepsilon_0$ и $(1-\frac{1}{\beta^t})\cdot\beta^u = \varepsilon_\infty$

```
     МАНТИССА      : ПОРЯДОК:          ЧИСЛО
--------------------+--------+-------------------------
 .5000000000000000 :      1 :   .100000000000000E+001
 .5000000000000000 :    -51 :   .222044604925031E-015
 .5000000000000000 :  -1021 :   .222507385850720E-307
 .9999999999999999 :   1024 :   .179769313486232E+309
```

Пример 2. Восстановить (получить) с помощью программы INIT константы β, l, u, t и $\varepsilon_1, \varepsilon_0, \varepsilon_\infty$ следующих компьютеров: Convex 120, Convex 240, СМ-4, ЕС ЭВМ, IBM PC, Spp 2000, Sun, Vax ЭВМ.

Решение такой задачи получается с помощью программы INIT, если ее использовать на компьютерах указанных типов.

Результаты работы программы INIT

В таблицах 4 и 5 приведены указанные константы вещественной арифметики соответствующих ЭВМ, полученные с помощью программы INIT. При этом в таблицах 4 и 5 вещественные числа $\varepsilon_\infty, \varepsilon_0, \varepsilon_1, \varepsilon_2$ представлены в десятичной системе счисления.

Таблица 4

Тип ЭВМ	β	l	u	t	$\varepsilon_2 = \varepsilon_1/\beta$
Convex 120, Sun, Spp 2000	2	-1021	1024	53	1.110223024625157E-16
Convex 240	2	-1023	1023	53	1.110223024625157E-16
Vax, СМ-4	2	-127	127	56	1.387778780781416E-17
IBM PC	2	-1021	1024	53	1.110223024625157E-16
ЕС ЭВМ	16	-64	63	14	1.387778780781416E-17

Таблица 5

Тип ЭВМ	ε_0	ε_∞	ε_1
Convex 120, Sun, Spp 2000	2.225073858507201E-308	1.797693134862316E308	2.220446049250313E-16
Convex 240	5.562684646268003E-309	8.988465674311579E308	2.220446049250313E-16
Vax, СМ-4	2.938735877055719E-039	1.701411834604692E038	2.775557561562891E-17
IBM PC	2.225073858507201E-308	1.797693134862316E308	2.220446049250313E-16
ЕС ЭВМ	5.397605346934028E-079	7.237005577332262E075	2.220446049250313E-16

2. Описание программ LinsysccmSolver, Lin2dsysccmSolver, Lin3dsysccmSolver и PseudsysccmSolver — решения систем линейных алгебраических уравнений $C_2X = Y, C_3X = Y$ и $AX = Y$. Примеры тестовых расчетов

В этом параграфе приводится описание программ:

Lin2dsysccmSolver — обращения матриц C_2 и решения систем линейных уравнений $C_2X = Y$;

Lin3dsysccmSolver — обращения матриц C_3 и решения систем линейных уравнений $C_3X = Y$;

LinsysccmSolver — обращения матриц A и решения систем линейных уравнений $AX = Y$;

PseudsysccmSolver — решения систем линейных уравнений $AX = Y$.

Здесь $A[m,n] = (a_{ij})$ — вещественная матрица общего вида,

$$C_3 = \begin{bmatrix} q_1 & r_2 & & 0 \\ p_2 & q_2 & \ddots & \\ & \ddots & \ddots & r_m \\ 0 & & p_m & q_m \end{bmatrix}, \quad C_2 = \begin{bmatrix} q_1 & r_2 & & 0 \\ & q_2 & \ddots & \\ & & \ddots & r_m \\ 0 & & & q_m \end{bmatrix}. \qquad (2.1)$$

Приводятся также некоторые результаты тестовых расчетов.

Описание программ Lin2dsysccmSolver, Lin3dsysccmSolver, LinsysccmSolver и PseudsysccmSolver

Программа Lin2dsysccmSolver

Структура: SUBROUTINE
Внешние подпрограммы и функции: нет.
Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
Обращение: CALL Lin2dsysccmSolver(M,R,N,INF,IM,JM,B,DET)
Входные данные: блок RCONST содержит вещественные константы ε_∞, $\varepsilon_0, \varepsilon_1, \varepsilon_2 = \varepsilon_1/\beta$, определенные в INIT (или заранее известные);
M – (integer) порядок квадратной матрицы C_2 (2.1);
N – (integer) первая размерность массива B (N=M);
R – (real*8) одномерный (или двумерный) массив размерности 2M (или [2,M]). На входе в массиве R размещается исходная матрица C_2 в виде

$$R = [*, q_1, r_2, q_2, ..., r_{M-1}, q_{M-1}, r_M, q_M] \text{ или } R = \begin{bmatrix} * & r_2 & \cdots & r_M \\ q_1 & q_2 & \cdots & q_M \end{bmatrix}. \qquad (2.2)$$

Здесь $\{r_i\}_{i=2}^m$ — внедиагональные элементы матрицы C_2, символом $*$ отмечены ячейки, которые при вводе не заполняются;
INF – (integer): INF= 1, если задача решается с верхнедвухдиагональной матрицей; INF= −1, если задача решается с нижнедвухдиагональной матрицей;
IM,JM – (integer). Если JM=0, то решается система $C_2X = Y$. При этом IM — число правых частей. Если же JM= k ($k \le$IM), то вычисляются элементы обратной матрицы C_2^{-1}, находящиеся в столбцах с номерами $k, k+1, ...,$IM;
B – (real*8) двумерный*) массив. Если решается система $C_2X = Y$, т.е. JM=0, то массив B имеет размерность [N,IM]. При этом в B хранится матрица правых частей в виде

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1IM} \\ b_{21} & b_{22} & \cdots & b_{2IM} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NIM} \end{bmatrix}. \qquad (2.3)$$

Если вычисляются элементы матрицы C_2^{-1}, т.е. JM$\neq$0, то массив имеет размерность [N,IM-JM+1];
DET – (real*8) если DET=1, то на выходе DET= $\det(C_2)$ — определитель матрицы C_2, если DET=0, то вычисления определителя не производится.
Выходные данные: INF – (integer): INF= 0 – нормальное завершение работы подпрограмм; INF= −1 – исходная матрица вырождена;
B – (real*8) массив содержит матрицу решений и элементов обратной матрицы в виде (2.3);
DET – (real*8) содержит вычисленный определитель матрицы C_2.

*) Если отведенная память компьютера не достаточна для хранения массива B размерности [N,IM], то обращение к подпрограмме Lin2dsysccmSolver организуется самим пользователем в цикле с соответствующим выделением из B блоков нужных размеров. В частности, B может быть одномерным, если IM= 1.

Метод: метод критических компонент, изложенный в [4].

Программа Lin3dsysccmSolver

Структура: SUBROUTINE
Внешние подпрограммы и функции: RL – функция вычисления величин Φ_i (см. $(1)^0$ – $(5)^0$ [4]).
Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
Обращение: CALL Lin3dsysccmSolver(M,R,N,INF,IM,JM,B,A,DET)
Входные данные: блок RCONST содержит вещественные константы ε_∞, $\varepsilon_0, \varepsilon_1, \varepsilon_2 = \varepsilon_1/\beta$, определенные в INIT (или заранее известные);
M – (integer) порядок квадратной матрицы C_3 (2.2);
N – (integer) первая размерность массива B (N=M);
R – (real*8) одномерный (или двумерный) массив размерности 3M (или [3,M]). На входе в массиве R размещается исходная матрица C_3, если она несимметричная:

$$R = [*, q_1, p_2, r_2, q_2, ..., p_M, r_M, q_M, *] \text{ или } R = \begin{bmatrix} * & r_2 & \cdots & r_M \\ q_1 & q_2 & \cdots & q_M \\ p_2 & \cdots & p_M & * \end{bmatrix}. \qquad (2.4)$$

Если C_3 — симметричная матрица, то в виде (2.2). Здесь символом $*$ отмечены ячейки, которые при вводе не заполняются;
INF – (integer): INF= 1, если задача решается с симметричной трехдиагональной матрицей; INF= −1, если задача решается с несимметричной трехдиагональной матрицей;
IM,JM – (integer). Если JM=0, то решается система $C_3X = Y$. При этом IM — число правых частей. Если же JM= k ($k \le$IM), то вычисляются элементы обратной матрицы C_3^{-1}, находящиеся в столбцах с номером $k, k+1, ...,$IM;
B – (real*8) двумерный массив. Если решается система $C_3X = Y$, т.е. JM=0, то массив B имеет размерность [N,IM]. При этом в B хранится матрица правых частей в виде (2.3). Если вычисляются элементы матрицы C_3^{-1}, т.е. JM$\neq$0, то массив имеет размерность [N,IM-JM+1];
A – (real*8) рабочий массив размерности [3,M];
DET – (real*8). Если DET=1, то на выходе DET= $\det(C_3)$ — определитель матрицы C_3, если DET=0, то вычисления определителя не производится.
Выходные данные: INF – (integer): INF= 0 – нормальное завершение работы подпрограмм; INF= −1 – исходная матрица вырождена;
B – (real*8) массив содержит матрицу решений и элементов обратной матрицы в виде (2.3);
DET – (real*8) содержит вычисленный определитель матрицы C_3.
Метод: метод критических компонент, изложенный в [4].

Программа LinsysccmSolver

Структура: SUBROUTINE
Внешние подпрограммы и функции: CHEXP,GTEXP, MSCL, INPR, BTD, BTDBK, REFL, RFLD, Lin2dsysccmSolver, Lin3dsysccmSolver.
Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN

Обращение: CALL LinsysccmSolver(M,A,N,INF,IM,JM,B,R,DET)

Входные данные: блок RCONST содержит вещественные константы ε_∞, $\varepsilon_0, \varepsilon_1, \varepsilon_2 = \varepsilon_1/\beta$, определенные в INIT (или заранее известные);

INF – (integer):

INF= 0, если решается система с верхнедвухдиагональной матрицей;

INF= 1, если решается система с симметричной трехдиагональной $C_3 = C_3^T$ матрицей;

INF= −1, если решается система с несимметричной трехдиагональной $C_3 \neq C_3^T$ матрицей;

INF= 2, если решается (без масштабирования) система с заполненной $A \neq A^T$ матрицей;

INF= −2, если решается (с масштабированием) система с заполненной $A \neq A^T$ матрицей;

INF= 3, если решается (без масштабирования) система с заполненной $A = A^T$ матрицей;

INF= −3, если решается (с масштабированием) система с заполненной $A = A^T$ матрицей;

INF= 4, если решается система с нижнедвухдиагональной матрицей;

M,N – (integer) размерности прямоугольной матрицы A;

A – (real*8) двумерный массив*), который содержит исходную матрицу A. При этом матрицы, например A, запоминаются в виде

$$A = (a_{11}, a_{21}, ..., a_{M1}; a_{12}, a_{22}, ..., a_{M2}; a_{13}, ..., a_{MN}); \tag{2.5}$$

IM,JM – (integer). Если JM=0, то решается система $AX = Y$. При этом IM — число правых частей. Если же JM= k ($k \leq$IM), то вычисляются элементы обратной матрицы A^{-1}, находящиеся в столбцах с номером $k, k+1, ...,$IM;

B – (real*8) двумерный массив. Если решается система $AX = Y$, т.е. JM=0, то массив B имеет размерность [max(M,N),IM]. При этом в B хранится матрица правых частей в виде (2.5). Если вычисляются элементы матрицы A^{-1}, т.е. JM≠0, то массив имеет размерность [N,IM-JM+1];

R – (real*8) одномерный рабочий массив размерности 2M или 3M. В нем на входе содержится (при INF= 0; 4; 1; −1) двухдиагональная или симметричная трехдиагональная матрица в виде (2.2), а несимметричная трехдиагональная матрица в виде (2.4) соответственно;

DET – (real*8) если DET=1, то на выходе DET= $\det(A)$ — определитель матрицы A, если DET=0, то вычисления определителя не производится.

Выходные данные: INF – (integer): INF= 0 – нормальное завершение работы подпрограмм; INF= −1 – исходная матрица вырождена;

A – (real*8) массив содержит матрицы P и Q [10];

B – (real*8) массив содержит матрицу решений и элементов обратной матрицы в виде (2.5);

DET – (real*8) содержит вычисленный определитель матрицы A.

Метод: метод критических компонент, изложенный в [4].

*)Размерность массива A: [M,N], если на входе |INF| > 1; [M,3], если на входе INF= 1; [1,1] (или A – вещественная переменная), если на входе INF=0.

Программа PseudsysccmSolver

Структура: SUBROUTINE

Внешние подпрограммы и функции: CHEXP, GTEXP, MSCL, INPR, BTD, BTDBK, REFL, RFLD, Lin2dsysccmSolver, Lin3dsysccmSolver.

Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN

Обращение: CALL PseudsysccmSolver(M,A,N,INF,IM,B,R,C,P)

Входные данные: блок RCONST содержит вещественные константы ε_∞, $\varepsilon_0, \varepsilon_1, \varepsilon_2 = \varepsilon_1/\beta$, определенные в INIT (или заранее известные);

M,N – (integer) размерности прямоугольной матрицы A;

A – (real*8) двумерный массив, который содержит исходную матрицу A. При этом матрицы, например A, запоминаются в виде (2.5);

INF – (integer):

INF= 1, если решается система с верхнедвухдиагональной матрицей;

INF= −1, если решается система с нижнедвухдиагональной матрицей;

INF= 2, если решается (без масштабирования) система с заполненной A матрицей;

INF= −2, если решается (с масштабированием) система с заполненной A матрицей;

IM – (integer) число правых частей;

B – (real*8) двумерный массив. Имеет размерность [max(M,N),IM]. При этом в B хранится матрица правых частей в виде (2.5);

R – (real*8) одномерный рабочий массив размерности 2M. В нем на входе содержится (при INF= −1 и INF=1) двухдиагональная матрица в виде (2.2);

C,P – (real*8) рабочие массивы размерности 2M и 3M соответственно.

Выходные данные: INF – (integer): INF= 0 – нормальное завершение работы подпрограмм; INF= −1 – исходная матрица вырождена;

A – (real*8) массив содержит матрицы P и Q [10];

B – (real*8) массив содержит матрицу решений в виде (2.5).

Метод: метод критических компонент [4].

Пример *(тест)* **3.** Вычислить решения системы [1] (см. также приложение) 1, 10, 17 с $m = 3$, а также систем [11]:

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 2 \end{bmatrix} X = \begin{bmatrix} 2 \\ 1 \end{bmatrix}. \tag{2.6}$$

Вычислить обратные матрицы и определители матриц систем 1, 10, 17.

Ниже приводятся текст программы решения этой задачи, а также таблицы 6 — результаты работы программы.

```
PROGRAM TEST
IMPLICIT REAL*8 (A-H, O-Z)
INTEGER RADIX
DIMENSION A(10,10),XB(10,5),R(15),X(10)
COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
```

```
      CALL INIT
      M=3
      N=10
      DET=1D0
      JM=-1
      IM=1
    2 JM=JM+1
      INF=1
      DO I=1,3
      R(2*I)=1D0
      R(2*I-1)=2D0
      X(I)=XB(I,1)
      ENDDO
      XB(1,1)=2D0
      XB(2,1)=7D0/6D0
      XB(3,1)=1D0/3D0
      CALL Lin2dsysccmSolver(M,R,N,INF,IM,JM,XB,DET)
      IM=M
      IF(JM.EQ.0) GOTO 2
      PRINT 7
    7 FORMAT('Решение системы 1 и обращение (двухдиагональной)'
     *' матрицы этой системы:'/'       (Программа Lin2dsysccmSolver)'/
     *'  Решение              Обратная матрица')
      write(6,5) (X(I),(XB(I,J),J=1,IM),I=1,M)
      write(6,6) INF,DET
      det=1.
      JM=-1
      IM=1
    3 JM=JM+1
      inf=-1
      DO I=1,3
      R(3*I)=3D0
      R(3*I-1)=6D0
      R(3*I-2)=4D0
      X(I)=XB(I,1)
      ENDDO
      XB(1,1)=10D0
      XB(2,1)=13D0
      XB(3,1)=9D0
      CALL Lin3dsysccmSolver(M,R,N,INF,IM,JM,XB,A,DET)
      IM=M
      IF(JM.EQ.0) GOTO 3
      PRINT 8
    8 FORMAT('Решение системы 10 и обращение (трехдиагональной)'
     *' матрицы этой системы:'/'       (Программа Lin3dsysccmSolver)'/
     *'  Решение              Обратная матрица')
```



```
      write(6,5) (X(I),(XB(I,J),J=1,IM),I=1,M)
      write(6,6) INF,DET
      DO NUM=2,3
      det=1.
      JM=-1
      IM=1
    4 JM=JM+1
      IJ=0
      DO I=1,3
      DO J=1,3
      D=I+J-1
      A(IJ+J,1)=1D0/D
      ENDDO
      IJ=IJ+3
      X(I)=XB(I,1)
      ENDDO
      XB(1,1)=7D0/6D0
      XB(1,1)=XB(1,1)*XB(1,1)
      XB(2,1)=3D0/4D0
      XB(3,1)=21D0/40D0
      INF=NUM
      CALL LinsysccmSolver(M,A,M,INF,IM,JM,XB,R,DET)
      IM=M
      IF(JM.EQ.0) GOTO 4
      PRINT 9
    9 FORMAT('Решение системы 17 и обращение матрицы (общего'
     *' вида) этой системы:'/'       (Программа LinsysccmSolver)'/
     *'  Решение              Обратная матрица')
      write(6,5) (X(I),(XB(I,J),J=1,IM),I=1,M)
      write(6,6) INF,DET
      ENDDO
    5 FORMAT(D10.3,' : ',3d10.3)
    6 FORMAT(' INF=',I3,', DET=',d10.4,/,44('-'))
      DET=0.
      INF=2
      A(1,1)=1D0
      A(2,1)=2D0
      A(3,1)=1D0
      A(4,1)=-1D0
      A(5,1)=1D0
      A(6,1)=2D0
      XB(1,1)=2D0
      XB(2,1)=1D0
      CALL LinsysccmSolver(2,A,3,INF,1,0,XB,R,DET)
      PRINT*,'Решение системы (2.6):'
      write(6,'(3d10.3)') (XB(J,1),J=1,M)
```

```
      print*,'inf=',inf
      STOP
      END
```

Таблица 6 *решения систем 1,10,17 и обратных матриц к матрицам системы, а также решение системы* (2.5).

```
Решение системы 1 и обращение (двухдиагональной) матрицы этой системы
       (Программа Lin2dsysccmSolver)
  Решение                 Обратная матрица
 .100D+01 :   .100D+01 -.200D+01  .400D+01
 .500D+00 :   .000D+00  .100D+01 -.200D+01
 .333D+00 :   .000D+00  .000D+00  .100D+01
INF=  0, DET= .1000D+01
------------------------------------------
Решение системы 10 и обращение (трехдиагональной) матрицы этой системы
       (Программа Lin3dsysccmSolver)
  Решение                 Обратная матрица
 .100D+01 :   .333D+00 -.333D+00  .222D+00
 .100D+01 :  -.250D+00  .500D+00 -.333D+00
 .100D+01 :   .125D+00 -.250D+00  .333D+00
INF=  0, DET= .7200D+02
------------------------------------------
Решение системы 17 и обращение матрицы (общего вида) этой системы
       (Программа LinsysccmSolver)
  Решение                 Обратная матрица
 .100D+01 :   .900D+01 -.333D+00  .222D+00
 .500D+00 :  -.360D+02  .500D+00 -.333D+00
 .333D+00 :   .300D+02 -.250D+00  .333D+00
INF=  0, DET= .4630D-03
------------------------------------------
Решение системы 17 и обращение матрицы (общего вида) этой системы
       (Программа LinsysccmSolver)
  Решение                 Обратная матрица
 .100D+01 :   .900D+01 -.333D+00  .222D+00
 .500D+00 :  -.360D+02  .500D+00 -.333D+00
 .333D+00 :   .300D+02 -.250D+00  .333D+00
INF=  0, DET= .4630D-03
------------------------------------------
Решение системы (2.6):
 .500D+00  .100D+01  .500D+00
INF=          0
```

Заключение

Приведено описание машинно–независимого пакета программ `JINRLINPACK`, в котором реализованы алгоритмы, изложенные в [1–4;6], и который поставлен [5] на базовых ЭВМ ОИЯИ. В пакет входят программы:

`INIT` — получения констант вещественной арифметики ЭВМ;
`Lin2dsysccmSolver` — решения систем линейных уравнений $C_2X = Y$;
`Lin3dsysccmSolver` — решения систем линейных уравнений $C_3X = Y$;
`LinsysccmSolver`, `PseudsysccmSolver` — решения систем линейных уравнений $AX = Y$,

тексты которых приводятся ниже в приложение. Приведены некоторые результаты тестовых рассчетов, а также примеры использования программ.

Список литературы

[1] Emel'yanenko G.A., Rakhmonov T.T., Dushanov E.B. JINR preprint, E11-96-105, Dubna, 1996.

[2] Emel'yanenko G.A., Rakhmonov T.T., Dushanov E.B. JINR preprint, E11-96-106, Dubna, 1996.

[3] Emel'yanenko G.A., Rakhmonov T.T., Dushanov E.B. JINR preprint, E11-96-107, Dubna, 1996.

[4] Emel'yanenko G.A., Emelianenko M.G., Rakhmonov T.T., Dushanov E.B., Konovalova G.Yu. JINR preprint, E11-98-302, Dubna, 1998.

[5] Душанов Э.Б., Емельяненко М.Г., Коновалова Г.Ю. О форматах представления вещественных чисел и алгоритм автоматического определения констант вещественной арифметики ЭВМ. Препринт ОИЯИ P11-2000-163, Дубна, 2000.

[6] Новости ОИЯИ (JINR News). Информационный бюллетень ОИЯИ, 3, 1996, p.12.

[7] CERNLIB — CERN Program Library (Short Writeups). Application Software Group. Computing and Networks Division. CERN, Geneva, Switzerland (May, 1993).

[8] Райс Дж. Матричные вычисления и математическое обеспечение. М., "Мир", 1984.

[9] Федорова Р. Н., Широкова А. И. Библиотека программ на ФОРТРАНе. т. VI-VII. Описание программ. Дубна, 1990.

[10] Малышев А.Н. Введение в вычислительную линейную алгебру (с приложением алгоритмов на ФОРТРАНе). Новосибирск, "Наука", 1991.

[11] Фаддеева В.Н., Колотилина Л.Ю. Вычислительные методы линейной алгебры: Набор матриц для тестирования, ч.1,2,3 (Материалы по мат. обеспечению ЭВМ), Ленинград, 1982.

Приложение

Тестовые примеры и тексты программ LinsysccmSolver, Lin2dsysccmSolver, Lin3dsysccmSolver, PseudsysccmSolver, INIT, GTEXP, CHEXP машинно – независимого пакета JINRLINPACK на Фортране–77

I. Тестовые примеры систем уравнений $C_2X = Y$ с двухдиагональными матрицами общего вида.

Система 1. C_2 — матрица порядка m.

$$C_2 = \begin{bmatrix} 1 & 2 & & 0 \\ & \ddots & \ddots & \\ & & 1 & 2 \\ 0 & & & 1 \end{bmatrix}, \begin{cases} x_i = 1/i, \\ i = 1,2,...,m, \\ y_i = \frac{3i+1}{i(i+1)},\ y_m = 1/m, \\ i = 1,2,...,m-1. \end{cases}$$

Система 2. C_2 — матрица порядка $m, \varepsilon^* = 0.01$.

$$C_2 = \begin{bmatrix} \varepsilon^* & 1-\varepsilon^* & & 0 \\ & \ddots & \ddots & \\ & & \varepsilon^* & 1-\varepsilon^* \\ 0 & & & \varepsilon^* \end{bmatrix}, \begin{cases} x_i = 1/(2i+\varepsilon^*), i = 1,2,...,m, \\ y_i = \frac{2i+3\varepsilon^*}{(2i+\varepsilon^*)(2i+\varepsilon^*+2)}, \\ i = 1,2,...,m-1, \\ y_m = \varepsilon^*/(2m+\varepsilon^*). \end{cases}$$

Система 3. C_2 — матрица порядка m.

$$C_2 = \begin{bmatrix} \frac{7}{5} & \frac{11}{3} & & 0 \\ & \ddots & \ddots & \\ & & \frac{7}{5} & \frac{11}{3} \\ 0 & & & \frac{7}{5} \end{bmatrix}, \begin{cases} x_i = 1/(2i+1), \\ i = 1,2,...,M, \\ y_i = \frac{152i+118}{15(2i+1)(2i+3)},\ y_m = 7/5(2m+1), \\ i = 1,2,...,m-1. \end{cases}$$

Система 4. C_2 — матрица порядка $m, \varepsilon^* = 10^{-7}, \tilde{\varepsilon}^* = 10^{-4}$.

$$C_2 = \begin{bmatrix} \varepsilon^*\ 2 \\ -1\ 2 \\ \cdots \\ -1\ 2 \\ \tilde{\varepsilon}^*\ 2 \\ -1\ 2 \\ \cdots \\ -1\ 2 \\ \tilde{\varepsilon}^* \end{bmatrix}, \begin{cases} x_i = (-1)^{i+1}(1+\varepsilon^*), \\ i = 1,2,...,m, \\ y_1 = (\varepsilon^*-2)(1+\varepsilon^*), \\ y_i = (-1)^i 3(1+\varepsilon^*), \\ i = 2,3,...,k-1,k+1,...,m-1, \\ y_k = (-1)^k(2-\tilde{\varepsilon}^*)(1+\varepsilon^*), \\ y_m = (-1)^{m+1}\tilde{\varepsilon}^*(1+\varepsilon^*). \end{cases}$$

Система 5. C_2 — матрица порядка m.

$$C_2 = \begin{bmatrix} 3 & 7 & & 0 \\ & \ddots & \ddots & \\ & & 3 & 7 \\ 0 & & & 3 \end{bmatrix}, X = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}, Y = \begin{bmatrix} 10 \\ \vdots \\ 10 \\ 3 \end{bmatrix}.$$

II. Тестовые примеры систем линейных уравнений $C_3X = Y$ с трехдиагональными матрицами общего вида.

Система 6. C_3 — матрица порядка m.

$$C_3 = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix}, X = \begin{bmatrix} 1 \\ 1/2 \\ \vdots \\ 1/m \end{bmatrix}, y_i = \begin{cases} 3/2, & i = 1, \\ \frac{2}{(1-i)i(1+i)}, & i = 2,...,m-1, \\ \frac{m-2}{m(m-1)}, & i = m. \end{cases}$$

Система 7. C_3 — матрица порядка $m, \varepsilon^* = 10^{-7}$.

$$C_3 = \begin{bmatrix} -1 & 1 & & & 0 \\ 1 & -2 & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & \frac{1-m}{m} \end{bmatrix}, X = \begin{bmatrix} 1-\varepsilon^* \\ 1+\varepsilon^* \\ 1-\varepsilon^* \\ \vdots \\ 1+(-1)^m\varepsilon^* \end{bmatrix}, y_i = \begin{cases} 2\varepsilon^*, \quad i = 1, \\ (-1)^{i-1}4\varepsilon^*, \\ i = 2,...,m-1, \\ \frac{1+(-1)^m\varepsilon^*(1-2m)}{m}, \\ i = m. \end{cases}$$

Система 8. C_3 — матрица порядка m.

$$C_3 = \begin{bmatrix} 1 & -1 & & 0 \\ -1 & 1 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 1 \end{bmatrix}, X = \begin{bmatrix} 1/2 \\ 1/4 \\ \vdots \\ 1/2m \end{bmatrix}, y_i = \begin{cases} 1/4, & i = 1, \\ \frac{i^2+1}{2i(1-i)(1+i)}, & i = 2,...,m-1, \\ \frac{1}{2m(1-m)} & i = m. \end{cases}$$

Система 9. C_3 — матрица порядка $m, \varepsilon^* = 10^{-7}$.

$$C_3 = \begin{bmatrix} 1 & 1-\varepsilon^* & & & 0 \\ 1+\varepsilon^* & 1 & \ddots & & \\ & \ddots & \ddots & 1-\varepsilon^* & \\ & & 1+\varepsilon^* & 1 & 1-\varepsilon^* \\ 0 & & & 1+\varepsilon^* & 1 \end{bmatrix}, X = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}, Y = \begin{bmatrix} 2-\varepsilon^* \\ 3 \\ \vdots \\ 3 \\ 2+\varepsilon^* \end{bmatrix}.$$

Система 10. C_3 — матрица порядка m.

$$C_3 = \begin{bmatrix} 6 & 3 & & & 0 \\ 4 & 6 & \ddots & & \\ & \ddots & \ddots & 3 & \\ & & 4 & 6 & 3 \\ 0 & & & 4 & 6 \end{bmatrix}, X = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}, Y = \begin{bmatrix} 9 \\ 13 \\ \vdots \\ 13 \\ 10 \end{bmatrix}.$$

III. Тестовые примеры систем уравнений $AX = Y$ с $A \neq A^T$ - заполненными матрицами общего вида.

Система 11. A — матрица порядка $m, \varepsilon^* = 10^{-7}$.

$$A = \begin{bmatrix} m & m-1 & \cdots & 2 & 333 \\ m-1 & m-1 & m-2 & \cdots & 1 \\ \vdots & m-2 & m-2 & \cdots & 1 \\ 2 & \vdots & \vdots & \ddots & \vdots \\ \varepsilon^* & 1 & 1 & \cdots & 1 \end{bmatrix}, \begin{cases} x_i = 1/i, \ i = 1,2,\dots,m, \\ y_1 = \sum\limits_{k=1}^{m-1} \frac{m-k+1}{k} + \frac{333}{m}, y_m = \sum\limits_{k=2}^{m} \frac{1}{k} + \varepsilon^*; \\ y_i = (m-i+1)\sum\limits_{k=1}^{i} \frac{1}{k} + \sum\limits_{k=i+1}^{m} \frac{m-k+1}{k}, \\ i = 2,3,\dots,m-1. \end{cases}$$

Система 12. A — матрица порядка m.

$$\begin{cases} A = (a_{ij}), a_{ij} = \frac{1}{i+j-1}, \\ i = 1,\dots,m-1, j = 1,\dots,m, \\ a_{m1} = 333, \\ a_{mj} = \frac{1}{m+j-1}, j = 2,\dots,m, \end{cases} \quad \begin{cases} x_i = 1/(2i+1), i = 1,\dots,m, \\ y_i = \sum\limits_{k=1}^{m} \frac{1}{(2k+1)(i+k-1)}, i = 1,\dots,m-1, \\ y_m = \sum\limits_{k=2}^{m} \frac{1}{(2k+1)(m+k-1)} + 111. \end{cases}$$

Система 13. A — матрица порядка $m, \varepsilon^* = 10^{-5}$.

$$\begin{cases} A = (a_{ij}), a_{1j} = a_{j1} = \frac{1}{m-j+1}, \\ j = 1,2,\dots,m-1, \\ a_{1m} = 1+\varepsilon^*, \ a_{m1} = 1-\varepsilon^*, \\ a_{ij} = a_{ji} = \frac{1}{m-i+1}, \\ i = 2,3,\dots,m, j = 2,3,\dots,i, \\ x_i = 1-\varepsilon^*, i = 1,2,\dots,m, \end{cases} \quad \begin{cases} y_1 = (1-\varepsilon^*)(\sum\limits_{k=1}^{m-1} \frac{1}{m-k+1} + 1 + \varepsilon^*), \\ y_i = (1-\varepsilon^*)(\frac{i}{m-i+1} + \sum\limits_{k=i}^{m-1} \frac{1}{m-k}), \\ i = 2,3,\dots,m-1, \\ y_m = (1-\varepsilon^*)(1-\varepsilon^* + \sum\limits_{k=2}^{m} \frac{1}{m-k+1}). \end{cases}$$

Система 14. A — матрица порядка m.

$$A = \begin{bmatrix} m & m-1 & & 0 \\ m-1 & m-1 & \ddots & \\ \vdots & \vdots & \ddots & 1 \\ 1 & 1 & \cdots & 1 \end{bmatrix}, \begin{cases} x_i = (-1)^i/i, i = 1,\dots,m, \\ y_M = \sum\limits_{k=1}^{M} (-1)^k/k, \\ y_i = (m-i)\sum\limits_{k=1}^{i+1} \frac{(-1)^k}{k} + \sum\limits_{k=1}^{i} \frac{(-1)^k}{k}, \\ i = 1,\dots,m-1. \end{cases}$$

Система 15. A — матрица порядка m.

$$\begin{cases} A = (a_{ij}), a_{ij} = \frac{1}{i-j+m}, \\ i = 1,2,\dots,m, j = 1,2,\dots,m, \end{cases} \quad \begin{cases} x_i = 1/i, \\ y_i = \sum\limits_{k=1}^{m} \frac{1}{k(i-k+m)}, i = 1,\dots,m. \end{cases}$$

IV. Тестовые примеры систем уравнений $AX = Y$ с $A = A^T$ - заполненными матрицами общего вида.

Система 16. A — матрица порядка $m, \varepsilon^* = 10^{-7}$.

$$\begin{cases} A = (a_{ij}), a_{1j} = a_{j1} = \frac{1}{m-j+1}, \\ j = 1,2,\dots,m-1, \\ a_{m1} = a_{1m} = m+\varepsilon^*, \\ a_{ij} = a_{ji} = \frac{1}{m-i+1}, \\ i = 2,3,\dots,m, j = 2,3,\dots,i, \\ x_i = (i+1)/i, i = 1,2,\dots,m, \end{cases} \quad \begin{cases} y_1 = \sum\limits_{k=1}^{m-1} \frac{k+1}{k(m-k+1)} + \frac{(m+\varepsilon^*)(m+1)}{m}, \\ y_i = \frac{1}{m-i+1} \sum\limits_{k=1}^{i} \frac{k+1}{k} + \sum\limits_{k=i+1}^{m} \frac{k+1}{k(m-k+1)}, \\ y_m = \sum\limits_{k=2}^{m} \frac{k+1}{k} + 2(m+\varepsilon^*), \\ i = 1,2,\dots,m-1. \end{cases}$$

Система 17. A — матрица Гильберта порядка m.

$$\begin{cases} A = (a_{ij}), a_{ij} = \frac{1}{i+j-1}, \\ i = 1,2,\dots,m, j = 1,2,\dots,m, \end{cases} \quad \begin{cases} x_i = 1/i, i = 1,2,\dots,m, \\ y_i = \sum\limits_{k=1}^{m} \frac{1}{k(i+k-1)}, i = 1,2,\dots,m. \end{cases}$$

Система 18. A — матрица порядка m.

$$\begin{cases} A = (a_{ij}), a_{1j} = a_{j1} = \frac{1}{j}, \\ j = 1,2,\dots,m-1, \\ a_{m1} = a_{1m} = 333, a_{ij} = \frac{1}{i+j-1}, \\ i = 2,3,\dots,m, j = 2,3,\dots,m, \\ x_i = (-1)^i/i, i = 1,2,\dots,m, \end{cases} \quad \begin{cases} y_1 = \sum\limits_{k=1}^{m-1} \frac{(-1)^k}{k^2} + \frac{(-1)^M 333}{m}, \\ y_i = \sum\limits_{k=1}^{m} \frac{(-1)^k}{k(k+i-1)}, i = 2,4,\dots,m-1, \\ y_m = \sum\limits_{k=2}^{m} \frac{(-1)^k}{k(k+m-1)} - 333. \end{cases}$$

Система 19. A — матрица порядка $m, \varepsilon^* = 10^{-7}$.

$$A = \begin{bmatrix} \varepsilon^* & m-1 & \cdots & 2 & m \\ m-1 & m-1 & m-2 & \cdots & 1 \\ \vdots & m-2 & m-2 & \cdots & 1 \\ 2 & \vdots & \vdots & \ddots & \vdots \\ m & 1 & 1 & \cdots & 1 \end{bmatrix}, \begin{cases} x_i = 1-\varepsilon^*, i = 1,2,\dots,m, \\ y_1 = \frac{m(m+1)(1-\varepsilon^*)}{2} + (1-\varepsilon^*)^2, \\ y_i = \frac{(1-\varepsilon^*)(m+i)(m-i+1)}{2}, \\ y_m = (2m-1)(1-\varepsilon^*), \\ i = 2,3,\dots,m-1. \end{cases}$$

Система 20. A — матрица порядка $m, \det(A) = 0, \varepsilon^* = 10^{-11}, a = 1-\varepsilon^*, b = 1+\varepsilon^*$.

$$A = \begin{bmatrix} a & b & \cdots & b & a \\ b & a & b & \cdots & b \\ \vdots & b & \ddots & \vdots & \vdots \\ b & \vdots & \cdots & a & b \\ a & b & \cdots & b & a \end{bmatrix}, \begin{cases} x_i = \frac{(-1)^i}{2i+1}, i = 1,2,\dots,m, \\ y_1 = y_m = b\sum\limits_{k=2}^{m-1} \frac{(-1)^k}{2k+1} + a\left(\frac{(-1)^m}{2m+1} - \frac{1}{3}\right), \\ y_i = b\sum\limits_{k=1}^{m} \frac{(-1)^k}{2k+1} - \frac{(-1)^i 2\varepsilon^*}{2i+1}, \\ i = 2,3,\dots,m-1. \end{cases}$$

```
C
C                         Пакет программ JINRLINPACK
C
C Программа INIT - получения констант
C
C        beta - основание системы счисления,
C   (l<0,u>0) - нижняя и верхняя границы порядков представления
C               нормализованного числа в ЭВМ,
C           t - число beta-ичных разрядов, отведенных для хранения
C               мантиссы числа в ЭВМ,
C         e_8 - максимальное число, представимое в данной ЭВМ,
C         e_0 - минимальное число, представимое в данной ЭВМ,
C         e_1 - относительная погрешность вычислений с нормализованными
C               числами данной ЭВМ,
C
C а также получения параметров настройки функций GTEXP,CHEXP [5]
C
C Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C              COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
C              COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV
C Обращение:   CALL INIT, если печать вычисленных костант не требуется;
C              CALL INITP, если константы печатаются.
C Выходные данные: вещественные константы e_8,e_0,e_1,e_2=e_1/beta
C              и целые константы beta,u,l,t. Выдаются в блоках RCONST и
C              ICONST в указанных последовательностях соответственно.
C              В блоке ICHGT содержатся следующие выделенные параметры
C              настройки функции GTEXP, CHEXP:
C        BINV - (real*8) вещественное число равное 1/beta;
C          KB - (integer) номер половины формата, в котором хранится
C               младший разряд порядка нормализованного числа;
C         LWT - (integer) представление целого порядка e=0 в разрядах
C               формата, отведенных для порядка;
C         INO - (integer) количество beta-ичных разрядов, отведенных для
C               порядка с учетом его знака (в формате для хранения
C               вещественного числа);
C         INB - (integer) количество beta-ичных разрядов, отведенных для
C               мантиссы с учетом ее знака (в формате для хранения
C               вещественного числа);
C        MNE1 - (integer) признак кода (e>0)-положительного порядка.
C               При этом MNE1=[e]^k_(код)-e;
C        MNE2 - (integer) признак кода (e<0)-отрицательного порядка.
C               При этом MNE2=[e]^k_(код)-e.
C
      SUBROUTINE INIT
      IMPLICIT REAL*8 (A-H, O-Z)
      INTEGER RADIX,J1(2),J2(2),J3(2),N1(2),N2(2),N3(2)
      COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
      COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
      COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV
      EQUIVALENCE (J1,R1),(J2,R2),(J3,R3)
```



```
      EQUIVALENCE (OVFLO,N1),(UNDFLO,N2),(EPS,N3)
      GOTO 1
      ENTRY INITP
      LPCONST=1
    1 R1=1D0
      RADIX=1
      KB=1
C    ЦИКЛ ПОИСКА (BETA) - ОСНОВАНИЯ СИСТЕМЫ СЧИСЛЕНИЯ
    2 RADIX=RADIX+1
      R2=RADIX
      R3=1D0/R2
      ILT=J2(1)-J1(1)
      INO=J2(2)-J1(2)
      IF(ILT.NE.(J1(1)-J3(1)).OR.INO.NE.(J1(2)-J3(2))) GOTO 2
C    ЦИКЛ ПОИСКА t - РАЗРЯДОВ ОТВЕДЕННЫХ  ДЛЯ ХРАНЕНИИ МАНТИССЫ m
      D=R3*(R2-R1)
      UNDFLO=D
      MANTSZ=0
      BINV=R3
      IF(INO.EQ.0) GOTO 3
      ILT=INO
      KB=2
    3 OVFLO=UNDFLO
      MANTSZ=MANTSZ+1
      D=D*R3
      UNDFLO=UNDFLO+D
      EPS=UNDFLO
      IF(EPS.LT.R1) GOTO 3
      EPSMIN=R1-OVFLO
      MINEXP=2-MANTSZ
      EPS=EPSMIN
      UNDFLO=EPS*R2
      MNE2=N2(KB)
      IF(MNE2.LT.N3(KB)) MNE2=-MNE2
      MNE2=MNE2/ILT-MINEXP
      I=IABS(N3(KB)-N2(KB))
C    ЦИКЛ ВЫЧИСЛЕНИЯ e_0 - МИНИМАЛЬНО ПРЕДСТАВИМОГО
C    В ЭВМ ЧИСЛА И ЕГО ПОРЯДКА
    4 UNDFLO=EPS
      MINEXP=MINEXP-1
      D=UNDFLO*R3
      EPS=D
      IF(EPS.GT.ODO.AND.I.EQ.IABS(N3(KB)-N2(KB))) GOTO 4
C    ЦИКЛ ВЫЧИСЛЕНИЯ u - ПОРЯДКА МАКСИМАЛЬНО ПРЕДСТАВИМОГО
C    В ЭВМ ЧИСЛА e_8
      EPS=R1
      D=R2
      I=1
    5 MAXEXP=2*I
```

```
      INO=N3(KB).OR.INO
      IF(MAXEXP.GE.-MINEXP) GOTO 6
      EPS=EPS*D
      D=D*D
      I=MAXEXP
      GOTO 5
    6 IF((MAXEXP+MINEXP).LE.1) MAXEXP=MAXEXP-1
C     ЦИКЛ ВЫЧИСЛЕНИЯ e_8 - МАКСИМАЛЬНО ПРЕДСТАВИМОГО В ЭВМ ЧИСЛА
      OVFLO=OVFLO*EPS
      DO 7 K=I,MAXEXP
      OVFLO=OVFLO*R2
    7 CONTINUE
C     БЛОК ВЫЧИСЛЕНИЯ ПАРАМЕТРОВ ВЫДЕЛЕНИЯ ПОРЯДКА И МАНТИССЫ ЧИСЛА
      EPS=R3*R3
      I=MINO(N3(KB),N2(KB),J3(KB))
      INO=(INO-I).OR.(N2(KB)-I).OR.(J3(KB)-I)
      INO=INO.OR.(N3(KB)-I).OR.(J1(KB)-I)
      INB=.NOT.INO
      LWT=J3(KB).AND.INO
      MNE1=LWT/ILT
      EPS=EPSMIN*R2
      IF(LPCONST.NE.1) GOTO 8
      PRINT*,'The constants of mashin's arithmetic ...'
      PRINT9, MAXEXP,OVFLO,MINEXP,UNDFLO,RADIX,EPS,MANTSZ,EPSMIN
    9 FORMAT(1X,'MAXEXP=',I8,'   OVFLO=',1PE24.15E3,/,
     *       1X,'MINEXP=',I8,'  UNDFLO=',1PE24.15E3,/,
     *       1X,' RADIX=',I8,'     EPS=',1PE24.15E3,/,
     *       1X,'MANTSZ=',I8,'  EPSMIN=',1PE24.15E3)
    8 RETURN
      END
C
C Подпрограмма-функция GTEXP - выделения мантиссы и порядка вещественного
C                                числа
C Общие блоки: COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV
C Обращение:   RM=GTEXP(R,E)
C Входные данные: R - (real*8) вещественное число, порядок и мантисса
C                 которого выделяется. Блок ICHGT содержит настроечные
C                 параметры, определенные в INIT (или заранее известные);
C Выходные данные: RM - (real*8) содержит выделенную мантиссу числа R;
C                 E - (integer) содержит выделенный порядок числа R
C
      REAL FUNCTION GTEXP*8(R,E)
      INTEGER I(2),E
      REAL*8 R,Z,BINV
      COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV
      EQUIVALENCE (Z,I)
      IF(R.EQ.0D0) GOTO 2
      Z=R
      J=MNE1
```



```
      E=(I(KB).AND.INO)/ILT
      IF(DABS(Z).GE.BINV) GOTO 3
      IF(MNE2.LT.0) E=-E
      J=MNE2
    3 E=E-J
      I(KB)=(I(KB).AND.INB).OR.LWT
      GTEXP=Z
    1 RETURN
    2 GTEXP=0
      E=0
      GOTO 1
      END
C
C Подпрограмма-функция CHEXP - восстанавления вещественного числа по его
C                          мантиссе и порядку
C
C Общие блоки: COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
C              COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV
C Обращение:   R=CHEXP(RM,E)
C Входные данные: RM - (real*8) мантисса числа R; E - (integer) порядок
C              числа R. Блоки ICONST и ICHGT содержат элементы,
C              определенные в INIT (или заранее известные);
C Выходные данные: R - (real*8) содержит вещественное число beta^E RM=R
C
      REAL FUNCTION CHEXP*8(R,E)
      INTEGER I(2),L,J,E,RADIX
      REAL*8 R,Z,BINV
      COMMON /ICHGT/ INO,ILT,MNE1,MNE2,INB,LWT,KB,BINV
      COMMON /ICONST/ RADIX,MAXEXP,MINEXP,MANTSZ
      EQUIVALENCE (Z,I)
      IF(E.EQ.0.OR.R.EQ.0D0) GOTO 2
      Z=R
      K=MNE1
      J=(I(KB).AND.INO)/ILT
      IF(DABS(Z).GE.BINV) GOTO 4
      IF(MNE2.LT.0) J=-J
      K=MNE2
    4 L=J-K+E
      IF(L.GT.MAXEXP) L=MAXEXP
      K=MNE1
      IF(L.GT.0) GOTO 5
      IF(L.LT.MINEXP) GOTO 3
      K=IABS(MNE2)
      IF(MNE2.LT.0) L=-L
    5 I(KB)=(I(KB).AND.INB).OR.((L+K)*ILT)
      CHEXP=Z
    1 RETURN
    2 CHEXP=R
      GOTO 1
```

```
    3 CHEXP=0
      GOTO 1
      END
C
C Программа Lin2dsysccmSolver - обращения двухдиагональных матриц C_2 и
C                решения систем линейных уравнений C_2X=Y
C                   (метод критических компонент [1-4])
C
C Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C Обращение:   CALL Lin2dsysccmSolver(M,R,N,INF,IM,JM,B,DET)
C Входные данные: Блок RCONST содержит вещественные константы
C               e_8,e_0,e_1,e_2=e_1/beta, определенные в INIT
C               (или заранее известные);
C             M - (integer) порядок квадратной матрицы C_2;
C             N - (integer) первая размерность массива B (N=M);
C             R - (real*8) одномерный (или двумерный) массив
C                 размерности 2M. На входе в массиве R размещается
C                 исходная матрица C_2 в виде
C
C                  R=[*,q_1,r_2,q_2,...,r_{M-1},q_{M-1},r_M,q_M].    (*)
C
C                 Здесь {r_i}^m_{i=2} - внедиагональные элементы матрицы
C                 C_2, символом * отмечены ячейки, которые при вводе не
C                 заполняются;
C           INF - (integer): INF=1, если задача решается с
C                 верхнедвухдиагональной матрицей; INF=-1, если задача
C                 решается с нижнедвухдиагональной матрицей;
C         IM,JM - (integer). Если JM=0, то решается система C_2X=Y. При
C                 этом IM - число правых частей. Если же JM=k (k<=IM),
C                 то вычисляются элементы обратной к C_2 матрицы,
C                 находящиеся в столбцах с номерами k,k+1,...,IM. При
C                 этом k - номер последнего вычисляемого столбца
C                 обратной матрицы;
C             B - (real*8) двумерный массив (если отведенная память
C                 компьютера не достаточна для хранения массива B
C                 размерности [N,IM], то обращение к подпрограмме
C                 Lin2dsysccmSolver организуется самим пользователем в
C                 цикле с соответствующим выделением из B блоков
C                 нужных размеров. В частности, B может быть одномерным,
C                 если IM=1). Если решается система C_2X=Y, т.е. JM=0,
C                 то массив B имеет размерность [N,IM]. При этом в B
C                 хранятся векторы правых частей Y_j в виде
C
C                     Y_1    Y_2   ...  Y_{IM}
C                  _                         _
C                  | b_{11} b_{12} ... b_{1IM} |
C                  | b_{21} b_{22} ... b_{2IM} |
C                B=|   :      :     :     :    |.                 (**)
C                  | b_{N1} b_{N2} ... b_{NIM} |
```



```
C                  -                         -
C
C                 Если вычисляются элементы обратной к C_2 матрицы,
C                 т.е. JM!=0, то массив B имеет размерность [N,IM-JM+1];
C           DET - (real*8), если на входе DET=1, то на выходе
C                 DET=det(C_2) - определитель матрицы C_2, если на входе
C                 DET=0, то вычисления определителя не производится.
C Выходные данные: INF - (integer):
C                 INF=0 - нормальное завершение работы подпрограмм;
C                 INF=-1 - исходная матрица вырождена;
C             B - (real*8) массив содержит матрицу решений и элементов
C                 обратной матрицы в виде (**);
C           DET - (real*8) содержит вычисленный определитель матрицы C_2,
C                  если на входе DET=1
C
      SUBROUTINE Lin2dsysccmSolver(M,R,N,K1,IM,JM,B,DET)
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION R(2,*),B(N,*)
      COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
*
      IF(M.LE.0.OR.N.LE.0) PRINT100,M,N
      K=K1
      K1=-1
      L=(K+1)/2
      M2=M
      IF(K.EQ.-1) M2=1
      M1=M-M2+1
      Q1=2D0*EPS
      IF(DET.EQ.0D0) GOTO 2
      DET=1D0
      IE=0
      DO 1 I=M1,M2,K
      DET1=GTEXP(R(2,I),JE)
      DET=DET*DET1
      DET=GTEXP(DET,IJE)
      IE=IE+JE+IJE
    1 CONTINUE
      DET=CHEXP(DET,IE)
    2 IF(DABS(R(2,M2)).LT.UNDFLO) R(2,M2)=EPS
      IF(JM.NE.0) GOTO 10
      DO 9 J=1,IM
      I=M2
      B(M2,J)=B(M2,J)/R(2,M2)
    3 I=I-K
      IF(DABS(R(2,I)).LT.UNDFLO) RETURN
      D1=(B(I,J)-R(1,I+L)*B(I+K,J))/R(2,I)
      IF(I.EQ.M1) GOTO 8
      DB=DABS(B(I,J))
      IF(DABS(DABS(R(2,I)*D1+R(1,I+L)*B(I+K,J))-DB).GT.Q1) GOTO 4
```

```
      B(I,J)=D1
      GOTO 3
    4 D=R(1,I+L)*B(I+K,J)
      D0=B(I,J)
    5 D1=D0/R(2,I)-D/R(2,I)
      D2=D0/R(2,I)
      B(I,J)=D1
      IF(I.EQ.M1) GOTO 9
      I=I-K
      IF(DABS(R(2,I)).LT.UNDFLO) RETURN
      D0=B(I,J)-R(1,I+L)*D0/R(2,I+K)
      IF(DABS(D).GT.DABS(R(1,I+L))) GOTO 6
      D=-D/R(2,I+K)*R(1,I+L)
      GOTO 7
    6 D=-R(1,I+L)/R(2,I+K)*D
    7 IF(DABS(D).GE.1D0/EPS) GOTO 4
      IF(DABS(DABS(D0+R(1,I+L)*D2)-DABS(B(I,J))).GT.Q1) GOTO 4
      GOTO 5
    8 B(I,J)=D1
    9 CONTINUE
      GOTO 19
C
   10 J=0
      DO 18 J0=JM,IM
      J=J+1
      I=J0
      IF(DABS(R(2,I)).LT.UNDFLO) RETURN
      DO 11 L1=M2,I+K,-K
   11 B(L1,J)=0D0
      B(I,J)=1D0/R(2,I)
      IF(I.EQ.M1) GOTO 18
   12 I=I-K
      IF(DABS(R(2,I)).LT.UNDFLO) RETURN
      D1=-R(1,I+L)*B(I+K,J)/R(2,I)
      IF(I.EQ.M1) GOTO 17
      IF(DABS(R(2,I)*D1+R(1,I+L)*B(I+K,J)).GT.Q1) GOTO 13
      B(I,J)=D1
      GOTO 12
   13 D=R(1,I+L)*B(I+K,J)
   14 D1=-D/R(2,I)
      B(I,J)=D1
      IF(I.EQ.M1) GOTO 18
      I=I-K
      IF(DABS(R(2,I)).LT.UNDFLO) RETURN
      IF(DABS(D).GT.DABS(R(1,I+L))) GOTO 15
      D=D1*R(1,I+L)
      GOTO 16
   15 D=-R(1,I+L)/R(2,I+K)*D
   16 IF(DABS(D).GE.1D0/EPS) GOTO 13
```



```
      GOTO 14
   17 B(I,J)=D1
   18 CONTINUE
   19 K1=0
      RETURN
  100 FORMAT(' M OR N LESS THEN 0 ON PROGRAM Lin2dsysccmSolver: M=',
     * I2,' N=',I2)
      END
C
C Программа Lin3dsysccmSolver - обращения трехдиагональных матриц C_3 и
C              решения систем линейных уравнений C_3X=Y
C                 (метод критических компонент [1-4])
C
C Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C Обращение:   CALL Lin2dsysccmSolver(M,R,N,INF,IM,JM,B,A,DET)
C Входные данные: Блок RCONST содержит вещественные константы
C              e_8,e_0,e_1,e_2=e_1/beta, определенные в INIT
C              (или заранее известные);
C            M - (integer) порядок квадратной матрицы C_3;
C            N - (integer) первая размерность массива B (N=M);
C            R - (real*8) одномерный (или двумерный) массив
C                размерности 3M. На входе в массиве R размешается
C                исходная матрица C_3, если она несимметричная, в виде
C
C                 R=[*,q_1,p_2,r_2,q_2,...,p_M,r_M,q_M].          (***)
C
C                Если она симметричная, то в виде (*). Здесь символом *
C                отмечены ячейки, которые при вводе не заполняются;
C          INF - (integer): INF=1, если задача решается с симметричной
C                трехдиагональной матрицей; INF=-1, если задача решается
C                с несимметричной трехдиагональной матрицей;
C        IM,JM - (integer). Если JM=0, то решается система C_3X=Y. При
C                этом IM - число правых частей. Если же JM=k (k<=IM),
C                то вычисляются элементы обратной к C_3 матрицы,
C                находящиеся в столбцах с номерами k,k+1,...,IM. При
C                этом k - номер последнего вычисляемого столбца
C                обратной матрицы;
C            B - (real*8) двумерный массив (если отведенная память
C                компьютера не достаточна для хранения массива B
C                размерности [N,IM], то обращение к подпрограмме
C                Lin3dsysccmSolver организуется самим пользователем в
C                цикле с соответствующим выделением из B блоков
C                нужных размеров. В частности, B может быть одномерным,
C                если IM=1). Если решается система C_3X=Y, т.е. JM=0,
C                то массив B имеет размерность [N,IM]. При этом в B
C                хранятся векторы правых частей Y_j в виде (**).
C                Если вычисляются элементы обратной к C_3 матрицы,
C                т.е. JM!=0, то массив B имеет размерность [N,IM-JM+1];
C            A - (real*8) рабочий массив размерности [3,M];
```

```
C          DET - (real*8) если на входе DET=1, то на выходе
C                DET=det(C_3) - определитель матрицы C_3, если на входе
C                DET=0, то вычисления определителя не производится.
C Выходные данные: INF - (integer):
C                INF=0 - нормальное завершение работы подпрограмм;
C                INF=-1 - исходная матрица вырождена;
C            B - (real*8) массив содержит матрицу решений и элементов
C                обратной матрицы в виде (**);
C          DET - (real*8) содержит вычисленный определитель матрицы C_3,
C                если на входе DET=1
C
      SUBROUTINE Lin3dsysccmSolver(M,R,N,INF,IM,JM,B,A,DET)
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION R(*),B(N,*),A(N,*)
      COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
*
      IF(M.LE.0.OR.N.LE.0) PRINT100,M,N
      IS=1
      DET1=DET
      Q1=2D0*EPS
      IN1=IABS(INF)+1
      IQ=(5-ISIGN(1,INF))/2
      JQ=2-IQ
      INFO=INF
      INF=-1
   71 IS=-IS
      M2=M+1
      I=M2*(1-IS)/2+IS
      M2=M2-I
      IN=IN1-IS
      NQ=JQ+IS
      MQ=JQ*(1-IS)+IS
      IL=0
      I1=I
      A(I,IN)=R(IQ*I+JQ)
    1 KQ=IQ*I
      IF(I.EQ.M2) GOTO 4
      I=I+IS
      IF(DABS(A(I-IS,IN)).GE.UNDFLO) GOTO 2
      A(I,IN)=-R(KQ+MQ)*R(KQ+NQ)
      DET=A(I,IN)
      IF(DABS(DET).LT.UNDFLO) RETURN
      IL=1
      IF(I.EQ.M2) GOTO 69
      I=I+IS
      A(I,IN)=R(IQ*I+JQ)
      GOTO 1
    2 A(I-IS,IN1)=R(KQ+MQ)/A(I-IS,IN)
      IF(DABS(A(I-IS,IN1)).GT.1D0) IL=1
```



```
      IF(DABS(R(KQ+MQ)).GT.DABS(R(KQ+NQ))) GOTO 3
      A(I,IN)=R(IQ*I+JQ)-A(I-IS,IN1)*R(KQ+NQ)
      GOTO 1
    3 A(I,IN)=R(IQ*I+JQ)-R(KQ+NQ)/A(I-IS,IN)*R(KQ+MQ)
      GOTO 1
    4 DET=A(I,IN)
      IF(DABS(DET).LT.UNDFLO) A(I,IN)=EPS
      IF(IL.NE.0) GOTO 69
      IF(JM.NE.0) GOTO 8
C  SOLUTION WITH THE SWEEP METHOD
      DO 7 J=1,IM
      B(I1,J)=B(I1,J)/A(I1,IN)
      DO 5 L=I1+IS,I,IS
    5 B(L,J)=(B(L,J)-R(IQ*(L-IS)+NQ)*B(L-IS,J))/A(L,IN)
      DO 6 L=I-IS,I1,-IS
    6 B(L,J)=B(L,J)-A(L,IN1)*B(L+IS,J)
    7 CONTINUE
      GOTO 72
C  INVERSION WITH THE SWEEP METHOD
    8 J=0
      DO 12 J1=JM,IM
      J=J+1
      DO 9 L=I1,J1-IS,IS
    9 B(L,J)=0D0
      B(J1,J)=1D0/A(J1,IN)
      DO 10 L=J1+IS,I,IS
   10 B(L,J)=-R(IQ*(L-IS)+NQ)/A(L,IN)*B(L-IS,J)
      DO 11 L=I-IS,J1,-IS
   11 B(L,J)=B(L,J)-A(L,IN1)*B(L+IS,J)
      DO 12 L=J1-IS,I1,-IS
      B(L,J)=-A(L,IN1)*B(L+IS,J)
   12 CONTINUE
      GOTO 72
   69 IF(IS.EQ.-1) GOTO 71
C
      M1=M2-IS
      NI=IN1+IS
      NQ1=NQ
      NQ=JQ-IS
      MQ=JQ*(1+IS)-IS
      IF(JM.NE.0) GOTO 38
      DO 37 J=1,IM
      D=B(I1,J)
      D0=B(M2,J)
      DO 13 L=I1+IS,M2,IS
      A(L,IN1)=-R(IQ*(L-IS)+NQ1)*D
      D=A(L,IN1)
      IF(DABS(A(L-IS,IN)).LT.UNDFLO) GOTO 13
      A(L,IN1)=A(L,IN1)/A(L-IS,IN)
```

```
      D=B(L,J)+A(L,IN1)
   13 CONTINUE
      B(M2,J)=D/A(M2,IN)
      D2=DABS(D0)
      L=M2
      LK=M2
   14 KQ=L*IQ
      L=L-IS
      D0=-R(KQ+NQ)*D0
      IF(DABS(A(L+IS,NI)).LT.UNDFLO) GOTO 20
      D0=B(L,J)+D0/A(L+IS,NI)
      IF(L.EQ.I1) GOTO 36
      IF(DABS(A(L-IS,IN)).LT.DABS(A(L+IS,NI))) GOTO 16
      D1=A(L,IN)
      IF(L.EQ.M1) GOTO 15
      IF(DABS(A(L+2*IS,NI)).LT.UNDFLO) GOTO 19
   15 D1=D1-R(KQ+NQ)*R(KQ+MQ)/A(L+IS,NI)
      GOTO 19
   16 IF(DABS(A(L-IS,IN)).GE.UNDFLO) GOTO 17
      IF(DABS(A(L,IN)).LT.UNDFLO) RETURN
      D=A(L,IN1)/A(L,IN)
      GOTO 21
   17 D1=A(L,NI)
      IF(L.EQ.I1+IS) GOTO 18
      IF(DABS(A(L-2*IS,IN)).LT.UNDFLO) GOTO 19
   18 D1=D1-R(IQ*L+NQ)*R(IQ*L+MQ)/A(L-IS,IN)
   19 IF(DABS(D1).LT.UNDFLO) RETURN
      D=(D0+A(L,IN1))/D1
      GOTO 21
   20 IF(L.EQ.I1) GOTO 36
      IF(DABS(A(L,NI)).LT.UNDFLO) RETURN
      D=D0/A(L,NI)
   21 D3=0D0
      IF(L.NE.M1) D3=R(KQ+NQ+IQ)*B(L+2*IS,J)
      IF(RL(R(KQ+MQ)*D,R(KQ+JQ)*B(L+IS,J),D3,D2).LE.Q1) GOTO 35
   22 IF(DABS(A(L,IN)).GE.UNDFLO) GOTO 23
      B(L,J)=D
      IF(LK.NE.M2) B(L,J)=DD+D1
      IF(L.EQ.I1) GOTO 37
      L=L-IS
   23 D2=R(IQ*L+JQ)
      LK=L-IS
      D=-R(IQ*(L+IS)+NQ)/A(L,IN)*B(L+IS,J)
      D0=B(L,J)
      DD=(A(L,IN1)+D0)/A(L,IN)
      D1=D
   24 KQ=L*IQ
      A(L,IN1)=DD
      DD2=DABS(B(L,J))
```



```
      B(L,J)=DD+D1
      IF(L.EQ.I1) GOTO 37
      L=L-IS
      D4=D2
      D=-R(KQ+NQ)*D
      IF(DABS(A(L,IN)).GE.UNDFLO) D=D/A(L,IN)
      IF(DABS(D).GT.1D0/EPS) GOTO 22
      IF(DABS(D2).GE.UNDFLO) GOTO 25
      D2=R(KQ+MQ)
      IF(L.EQ.I1) GOTO 34
      IF(DABS(D2).LT.UNDFLO) RETURN
      DD=D0/D2
      D1=D
      DD3=0D0
      IF(L.NE.LK) DD3=R(KQ+NQ+IQ)*A(L+2*IS,IN1)
      IF(RL(R(KQ+MQ)*DD,R(KQ+JQ)*A(L+IS,IN1),DD3,DD2).GT.Q1) GOTO 22
      DD2=DABS(B(L,J))
      A(L,IN1)=DD
      B(L,J)=DD+D
      KQ=L*IQ
      L=L-IS
      D=-R(KQ+NQ)/A(L,IN)*D
      D0=B(L,J)-R(KQ+NQ)*DD
      D2=R(IQ*L+JQ)
      IF(L.EQ.I1) GOTO 34
      IF(DABS(D).GT.1D0/EPS) GOTO 22
      IF(DABS(A(L-IS,IN)).LT.UNDFLO) GOTO 29
      D1=A(L,IN)
      GOTO 32
   25 D0=B(L,J)-R(KQ+NQ)*D0/D2
      IF(DABS(R(KQ+NQ)).LT.DABS(R(KQ+MQ))) GOTO 26
      D3=R(KQ+MQ)/D2*R(KQ+NQ)
      GOTO 27
   26 D3=R(KQ+NQ)/D2*R(KQ+MQ)
   27 D2=R(IQ*L+JQ)-D3
      IF(L.EQ.I1) GOTO 34
      IF(DABS(A(L-IS,IN)).LT.DABS(D4)) GOTO 28
      D1=A(L,IN)-D3
      GOTO 32
   28 IF(DABS(A(L-IS,IN)).GE.UNDFLO) GOTO 30
   29 IF(DABS(A(L,IN)).LT.UNDFLO) RETURN
      DD=A(L,IN1)/A(L,IN)
      D1=0D0
      GOTO 33
   30 D1=D2
      IF(L.EQ.I1+IS) GOTO 31
      IF(DABS(A(L-2*IS,IN)).LT.UNDFLO) GOTO 32
   31 D1=D1-R(IQ*L+NQ)*R(IQ*L+MQ)/A(L-IS,IN)
   32 IF(DABS(D1).LT.UNDFLO) RETURN
```

```
      DD=(A(L,IN1)+D0)/D1
      D1=D
   33 DD3=0D0
      IF(L.NE.LK) DD3=R(KQ+NQ+IQ)*A(L+2*IS,IN1)
      IF(RL(R(KQ+MQ)*DD,R(KQ+JQ)*A(L+IS,IN1),DD3,DD2).GT.Q1) GOTO 22
      GOTO 24
   34 IF(DABS(D2).LT.UNDFLO) RETURN
      B(L,J)=D0/D2+D
      GOTO 37
   35 D2=DABS(B(L,J))
      B(L,J)=D
      IF (L.NE.I1) GOTO 14
   36 B(I1,J)=D0/A(I1,NI)
   37 CONTINUE
      GOTO 72
C
   38 J=0
      DO 68 J1=JM,IM
      J=J+1
      L=M2
      B(J1,J)=1D0
      IF(J1.EQ.I1) GOTO 40
      IF(DABS(A(J1-IS,IN)).GE.UNDFLO) GOTO 40
      DO 39 I=J1,M2,IS
   39 B(I,J)=0D0
      L=J1
      GOTO 42
   40 DO 41 I=J1+IS,M2,IS
      B(I,J)=-R(IQ*(I-IS)+NQ1)*B(I-IS,J)
      IF(DABS(A(I-IS,IN)).LT.UNDFLO) GOTO 41
      B(I,J)=B(I,J)/A(I-IS,IN)
   41 CONTINUE
      B(M2,J)=B(M2,J)/A(M2,IN)
   42 LK=M2
      D0=1D0
      LM=I1
      IF(J1.EQ.M2) GOTO 44
      IF(DABS(A(J1+IS,NI)).GE.UNDFLO) GOTO 44
      DO 43 I=I1,J1,IS
   43 B(I,J)=0D0
      LM=J1
      IF(L.EQ.LM) GOTO 68
   44 KQ=IQ*L
      L=L-IS
      D=0D0
      IF(L.GE.J1) GOTO 45
      D1=A(L,NI)
      B(L,J)=-R(KQ+NQ)*D0
      D0=B(L,J)
      IF(DABS(A(L+IS,NI)).LT.UNDFLO) GOTO 50
      B(L,J)=B(L,J)/A(L+IS,NI)
      D0=B(L,J)
   45 IF(L.EQ.I1) GOTO 67
      IF(DABS(A(L-IS,IN)).LT.DABS(A(L+IS,NI))) GOTO 47
      IF(DABS(A(L+IS,NI)).LT.UNDFLO) GOTO 66
      D1=A(L,IN)
      IF(L.EQ.M1) GOTO 46
      IF(DABS(A(L+2*IS,NI)).LT.UNDFLO) GOTO 50
   46 D1=D1-R(KQ+NQ)*R(KQ+MQ)/A(L+IS,NI)
      GOTO 50
   47 D1=A(L,IN)
      IF(DABS(A(L-IS,IN)).GE.UNDFLO) GOTO 51
      IF(L.GE.J1) GOTO 50
      GOTO 66
   51 D1=A(L,NI)
      IF(L.EQ.I1+IS) GOTO 49
      IF(DABS(A(L-2*IS,IN)).LT.UNDFLO) GOTO 50
   49 D1=D1-R(IQ*L+NQ)*R(IQ*L+MQ)/A(L-IS,IN)
   50 IF(DABS(D1).LT.UNDFLO) RETURN
      D=B(L,J)/D1
      D3=0D0
      IF(L.NE.M1) D3=R(KQ+NQ+IQ)*B(L+2*IS,J)
      D2=0D0
      IF(L+IS.EQ.J1) D2=1D0
      If(RL(R(KQ+MQ)*D,R(KQ+JQ)*B(L+IS,J),D3,D2).LE.Q1) GOTO 66
   52 IF(DABS(A(L,IN)).GE.UNDFLO) GOTO 53
      B(L,J)=D
      IF(LK.NE.M2) B(L,J)=DD+D1
      IF(L.EQ.LM) GOTO 68
      L=L-IS
   53 D2=R(IQ*L+JQ)
      DD1=0D0
      LK=L
      D=-R(KQ+NQ)/A(L,IN)*B(L+IS,J)
      DD=0D0
      IF(LK.GE.J1) DD=B(L,J)/A(L,IN)
   54 D1=D
   55 KQ=L*IQ
      D0=B(L,J)
      B(L,J)=DD+D1
      IF(L.EQ.LM) GOTO 68
      L=L-IS
      D4=D2
      D=-R(KQ+NQ)*D
      IF(DABS(A(L,IN)).GE.UNDFLO) D=D/A(L,IN)
      IF(LK.LT.J1) GOTO 54
      IF(DABS(D2).GE.UNDFLO) GOTO 56
      D2=R(KQ+MQ)
```

```
      IF(L.EQ.I1) GOTO 65
      IF(DABS(D2).LT.UNDFLO) RETURN
      DD2=DD1
      DD1=DD
      DD=0D0
      IF(L.LT.J1) DD=D0/D2
      DD3=DABS(R(KQ+MQ)*DD+R(KQ+JQ)*DD1+R(KQ+NQ+IQ)*DD2)
      IF(L+IS.EQ.J1) DD3=DABS(1D0-D3)
      IF(DD3.GT.Q1) GOTO 52
      D0=B(L,J)
      B(L,J)=DD+D
      KQ=L*IQ
      L=L-IS
      D=-R(KQ+NQ)/A(L,IN)*D
      IF(DABS(D).GT.1D0/EPS) GOTO 52
      IF(L.LT.J1) B(L,J)=-R(KQ+NQ)*DD
      D2=R(IQ*L+JQ)
      IF(L.EQ.I1) GOTO 65
      IF(DABS(A(L-IS,IN)).LT.UNDFLO) GOTO 60
      D1=A(L,IN)
      GOTO 63
   56 IF(L.LT.J1) B(L,J)=-R(KQ+NQ)*D0/D2
      IF(DABS(R(KQ+NQ)).LT.DABS(R(KQ+MQ))) GOTO 57
      D3=R(KQ+MQ)/D2*R(KQ+NQ)
      GOTO 58
   57 D3=R(KQ+NQ)/D2*R(KQ+MQ)
   58 D2=R(IQ*L+JQ)-D3
      IF(L.EQ.I1) GOTO 65
      IF(DABS(D).GT.1D0/EPS) GOTO 52
      IF(DABS(A(L-IS,IN)).LT.DABS(D4)) GOTO 59
      D1=A(L,IN)-D3
      GOTO 63
   59 IF(DABS(A(L-IS,IN)).GE.UNDFLO) GOTO 61
   60 DD2=DD1
      DD1=DD
      IF(DABS(A(L,IN)).LT.UNDFLO) RETURN
      DD=B(L,J)/A(L,IN)
      D1=0D0
      GOTO 64
   61 D1=D2
      IF(L.EQ.I1+IS) GOTO 62
      IF(DABS(A(L-2*IS,IN)).LT.UNDFLO) GOTO 63
   62 D1=D1-R(IQ*L+NQ)*R(IQ*L+MQ)/A(L-IS,IN)
   63 IF(DABS(D1).LT.UNDFLO) RETURN
      DD2=DD1
      DD1=DD
      DD=B(L,J)/D1
      D1=D
   64 DD3=DABS(R(KQ+MQ)*DD+R(KQ+JQ)*DD1+R(KQ+NQ+IQ)*DD2)
```



```
      IF(L+IS.EQ.J1) DD3=DABS(1D0-DD3)
      IF(DD3.LE.Q1) GOTO 55
      GOTO 52
   65 IF(DABS(D2).LT.UNDFLO) RETURN
      B(L,J)=B(L,J)/D2+D
      GOTO 68
   66 B(L,J)=D
      IF (L.GT.LM) GOTO 44
      GOTO 68
   67 B(L,J)=B(L,J)/A(L,NI)
   68 CONTINUE
   72 INF=0
      IF(DET1.EQ.0D0) DET=0D0
      IF(DABS(DET).LT.UNDFLO) RETURN
      DET=1D0
      IE=0
      DO 73 I=M2,I1,-IS
      IF(DABS(A(I,IN)).LT.UNDFLO) GOTO 73
      DET1=GTEXP(A(I,IN),JE)
      DET=DET*DET1
      DET=GTEXP(DET,IJE)
      IE=IE+JE+IJE
   73 CONTINUE
      DET=CHEXP(DET,IE)
      RETURN
  100 FORMAT(' M OR N LESS THEN 0 ON PROGRAM Lin3dsysccmSolver: M=',
     * I2,' N=',I2)
      END
C
C Программа LinsysccmSolver - обращения матриц A и решения систем
C    линейных уравнений AX=Y (метод критических компонент [1-4])
C
C Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C Обращение:    CALL LinsysccmSolver(M,A,N,INF,IM,JM,B,R,DET)
C Входные данные: Блок RCONST содержит вещественные константы
C               e_8,e_0,e_1,e_2=e_1/beta, определенные в INIT
C               (или заранее известные);
C           INF - (integer):
C                 INF=0, если решается система с верхнедвухдиагональной
C                 матрицей;
C                 INF=1, если решается система с симметричной
C                 трехдиагональной C_3=C^T_3 матрицей;
C                 INF=-1, если решается система с несимметричной
C                 трехдиагональной C_3!=C^T_3 матрицей;
C                 INF=2, если решается (без масштабирования) система с
C                 несимметричной заполненной A!=A^T матрицей;
C                 INF=-2, если решается (с масштабированием) система с
C                 несимметричной заполненной A!=A^T матрицей;
C                 INF=3, если решается (без масштабирования) система с
```

```
C                 симметричной заполненной A=A^T матрицей;
C                 INF=-3, если решается (с масштабированием) система с
C                 симметричной заполненной A=A^T матрицей;
C                 INF=4, если решается система с нижнедвухдиагональной
C                 матрицей;
C           M,N - (integer) размерности прямоугольной матрицы A
C             A - (real*8) двумерный массив (размерность массива A:
C                 [M,N], если на входе |INF|>1; [M,3], если на входе
C                 |INF|=1; [1,1] (или A - вещественная переменная), если
C                 на входе INF=0;4), который содержит исходную матрицу A.
C                 При этом матрицы A запоминаются в виде
C
C     A=(a_{11},...,a_{M1};a_{12},...,a_{M2};a_{13},...,a_{MN}); (****)
C
C         IM,JM - (integer). Если JM=0, то решается система AX=Y. При
C                 этом IM - число правых частей. Если же JM=k (k<=IM),
C                 то вычисляются элементы обратной к A матрицы,
C                 находящиеся в столбцах с номерами k,k+1,...,IM. При
C                 этом k - номер последнего вычисляемого столбца
C                 обратной матрицы;
C             B - (real*8) двумерный массив (если отведенная память
C                 компьютера не достаточна для хранения массива B
C                 размерности [max(M,N),IM], то обращение к подпрограмме
C                 LinsysccmSolver организуется самим пользователем в
C                 цикле с соответствующим выделением из B блоков
C                 нужных размеров. В частности, B может быть одномерным,
C                 если IM=1). Если решается система AX=Y, т.е. JM=0,
C                 то массив B имеет размерность [max(M,N),IM]. При этом
C                 в B хранятся векторы правых частей Y_j в виде (****).
C                 Если вычисляются элементы обратной к A матрицы,
C                 т.е. JM!=0, то массив B имеет размерность [N,IM-JM+1];
C             R - (real*8) одномерный рабочий массив размерности 2M или
C                 3M. В нем на входе содержится (при INF=0;4;1;-1)
C                 двухдиагональная или симметричная
C                 трехдиагональная матрица в виде (*), несимметричная
C                 трехдиагональная матрица в виде (***);
C           DET - (real*8), если на входе DET=1, то на выходе
C                 DET=det(A) - определитель матрицы A, если на входе
C                 DET=0, то вычисления определителя не производится.
C Выходные данные: INF - (integer):
C                 INF=0 - нормальное завершение работы подпрограмм;
C                 INF=-1 - исходная матрица вырождена.
C             A - (real*8) массив содержит матрицы P и Q [10] в
C                 в разложении PAQ=C_2 или Q^TAQ=C_3;
C             B - (real*8) массив содержит матрицу решений и элементов
C                 обратной матрицы в виде (****);
C           DET - (real*8) содержит вычисленный определитель матрицы A,
C                 если на входе DET=1
C
```



```
      SUBROUTINE LinsysccmSolver(M,A,N,INF,IM,JM,B,R,DET)
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION A(M,*),B(*),R(*)
      INTEGER EAB,EX,MSCL
      COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C
      IF(M.LE.0.OR.N.LE.0) PRINT100,M,N
      MO=INF
      IMO=IABS(MO)
      JMO=JM
      MXM=MAXO(M,N)
      MNM=MINO(M,N)
      IF(MXM.LE.0) RETURN
      I=IM
      IF(IMO.EQ.1) GOTO 6
      INF=-1
      IF(MO.EQ.0.OR.MO.EQ.4) GOTO 7
      IF(JM.NE.0) I=IM-JM+1
      IF(MO.GT.0) GOTO 1
      IF(MSCL(0,M,N,A,EAB).EQ.0) RETURN
    1 CALL BTD(IMO,M,N,A,R)
      DM12=A(MNM-1,MNM-2)
      DM2=A(MNM,MNM-2)
      DM1=A(MNM,MNM-1)
      EX=0
      IF(MO.GT.0) GOTO 4
      IF(MSCL(0,2,M,R,EX).EQ.0) RETURN
      EAB=EAB+EX
      IF(JM.NE.0) GOTO 3
      IF(MSCL(0,MXM,IM,B,EX).EQ.0) CONTINUE
      EX=EX-EAB
    4 IF(JM.EQ.0) GOTO 2
    3 EX=-EAB
      JMO=0
      J=JM
      DO 11 K=1,I
      DO 10 L=1,M
   10 B((K-1)*M+L)=0DO
      B((K-1)*M+J)=1DO
   11 J=J+1
    2 CALL BTDBK(IMO,M,N,A,1,1,I,B)
      IF(IMO.EQ.2) GOTO 7
      INF=M-2
C
    6 IF(MNM.GT.2) GOTO 60
      IF(MNM.EQ.2) GOTO 70
      IF(DET.NE.0DO) DET=R(2)
      IF(DABS(R(2)).LT.UNDFLO) R(2)=EPS
      B(1)=B(1)/R(2)
```

```
      GOTO 15
70    IQ=5-ISIGN(1,INF)
      INF=-1
      JQ=2-IQ/2
      KQ=JQ-1
      IF(JM.EQ.0) GOTO 68
      J=0
      DO 69 L=JM,I
      J=J+1
      DO 69 K=1,2
      B((J-1)*M+K)=0D0
      IF(K.EQ.L) B((J-1)*M+K)=1D0
 69   CONTINUE
 68   IF(R(2).NE.0D0.AND.R(IQ+JQ).NE.0D0) GOTO 61
      DDET=-R(3)*R(IQ+KQ)
      IF(DET.NE.0D0) DET=DDET
      IF(DABS(DDET).LT.UNDFLO) DDET=EPS
      DO 62 K=1,I
      BK1=B((K-1)*M+1)
      B((K-1)*M+1)=R(IQ+JQ)*BK1/DDET-B((K-1)*M+2)/R(3)
      B((K-1)*M+2)=R(2)*B((K-1)*M+2)/DDET-BK1/R(IQ+KQ)
 62   CONTINUE
      GOTO 8
 61   DR=-R(IQ+KQ)/R(2)
      DP=-R(3)/R(IQ+JQ)
      IF(DABS(R(3)).GT.DABS(R(IQ+KQ))) GOTO 63
      DLM=R(IQ+JQ)-R(3)/R(2)*R(IQ+KQ)
      DGM=R(2)+DP*R(IQ+KQ)
      GOTO 64
 63   DLM=R(IQ+JQ)+DR*R(3)
      DGM=R(2)-R(IQ+KQ)/R(IQ+JQ)*R(3)
 64   IF(DET.NE.0D0) DET=DLM*R(2)
      IF(DABS(DP).GT.DABS(DR)) GOTO 65
      IF(DABS(DGM).LT.UNDFLO) DGM=EPS
      DO 66 K=1,I
      B((K-1)*M+2)=B((K-1)*M+2)/R(IQ+JQ)
      B((K-1)*M+1)=(B((K-1)*M+1)-R(IQ+KQ)*B((K-1)*M+2))/DGM
      B((K-1)*M+2)=B((K-1)*M+2)+DP*B((K-1)*M+1)
 66   CONTINUE
      GOTO 8
 65   DO 67 K=1,I
      IF(DABS(DLM).LT.UNDFLO) DLM=EPS
      B((K-1)*M+1)=B((K-1)*M+1)/R(2)
      B((K-1)*M+2)=(B((K-1)*M+2)-R(3)*B((K-1)*M+1))/DLM
      B((K-1)*M+1)=B((K-1)*M+1)+DR*B((K-1)*M+2)
 67   CONTINUE
      GOTO 8
 60   CALL Lin3dsysccmsolver(MNM,R,M,INF,I,JM0,B,A,DET)
      M1=3
```



```
      IF(IM0.EQ.3) GOTO 8
      RETURN
    7 IF ((M0.EQ.0).OR.((M0.EQ.2).AND.(M.GE.N))) INF=1
      CALL Lin2dsysccmSolver(MNM,R,M,INF,I,JM0,B,DET)
      M1=4
      IF(M0.EQ.0) GOTO 9
      DET=-DET
    8 A(MNM-1,MNM-2)=DM12
      A(MNM,MNM-1)=DM1
      A(MNM,MNM-2)=DM2
      IF(M.EQ.N) GOTO 15
      NI=N*(I-1)
      IF(M.GT.N) THEN
      L=1
      DO 12 K=N,NI,N
      L=L+1
      DO 12 J=1,N
   12 B(K+J)=B((L-1)*M+J)
      ELSE
      L=I+1
      DO 13 K=NI,N,-N
      L=L-1
      DO 13 J=M,1,-1
      B(K+J)=B((L-1)*M+J)
   13 B((L-1)*M+J)=0D0
      DO 14 J=M+1,N
   14 B(NI+J)=0D0
      ENDIF
   15 CALL BTDBK(M1,M,N,A,-1,1,I,B)
      IF(M0.GT.0) GOTO 9
      IF(MSCL(1,MXM,I,B,EX).EQ.0) CONTINUE
    9 INF=0
      RETURN
  100 FORMAT(' M OR N LESS THEN 0 ON PROGRAM LinsysccmSolver: M=',
     * I2,' N=',I2)
      END
C
C Программа PseudsysccmSolver - решения систем линейных уравнений AX=Y
C со ''слабо'' вырожденной (у A имеется лишь одно нулевое собственное
C       значение) матрицей A (метод критических компонент [1-4])
C
C Общие блоки: COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C Обращение:   CALL PseudsysccmSolver(M,A,N,INF,IM,B,R,C,P)
C Входные данные: блок RCONST содержит вещественные константы
C                 e_8,e_0,e_1,e_2=e_1/beta, определенные в INIT
C                 (или заранее известные)
C             M,N - (integer) размерности прямоугольной матрицы A;
C             INF - (integer):
C                   INF=1, если решается система с верхнедвухдиагональной
```

```
C                 матрицей;
C                 INF=-1, если решается система с нижнедвухдиагональной
C                 матрицей;
C                 INF=2, если решается (без масштабирования) система с
C                 заполненной A матрицей;
C                 INF=-2, если решается (с масштабированием) система с
C                 заполненной A матрицей;
C             A - (real*8) двумерный массив (размерность массива A:
C                 [M,N], если на входе |INF|=2; [1,1] (или A -
C                 вещественная переменная), если на входе |INF|=1),
C                 который содержит исходную матрицу A. При этом матрицы
C                 A запоминаются в виде (****);
C            IM - (integer) число правых частей;
C             B - (real*8) двумерный массив. Имеет размерность
C                 [max(M,N),IM]. При этом в B хранится матрица правых
C                 частей в виде (****);
C             R - (real*8) одномерный рабочий массив размерности 2M.
C                 В нем на входе содержится (при INF=-1 и INF=1)
C                 двухдиагональная матрица в виде (*);
C           C,P - (real*8) рабочие массивы размерности 2M и 3M
C                 соответственно.
C Выходные данные: INF - (integer):
C                 INF=0 - нормальное завершение работы подпрограмм;
C                 INF=-1 - исходная матрица вырождена;
C             A - (real*8) массив содержит матрицы P и Q [10] в
C                 в разложении PAQ=C_2;
C             B - (real*8) массив содержит матрицу решений в виде (****)
C
      SUBROUTINE PseudsysccmSolver(M,A,N,INF,IM,B,R,C,P)
      IMPLICIT REAL*8 (A - H, O - Z)
      DIMENSION A(M,*),B(*),R(*),C(*),P(*)
      INTEGER EAB,EX,MSCL
      COMMON /RCONST/ OVFLO,UNDFLO,EPS,EPSMIN
C
      IF(M.LE.0.OR.N.LE.0) PRINT100,M,N
      MO=INF
      IMO=IABS(MO)
      MXM=MAXO(M,N)
      MNM=MINO(M,N)
      M1=MNM-1
      IF(MXM.LE.0) RETURN
      INF=-1
      IF(IMO.EQ.1) GOTO 3
*
      IF(MO.GT.0) GOTO 1
      IF(MSCL(0,M,N,A,EAB).EQ.0) RETURN
    1 CALL BTD(IMO,M,N,A,R)
      EX=0
      IF(MO.GT.0) GOTO 2
```



```
      IF(MSCL(0,2,M,R,EX).EQ.0) RETURN
      EAB=EAB+EX
      IF(MSCL(0,MXM,IM,B,EX).EQ.0) CONTINUE
      EX=EX-EAB
    2 CALL BTDBK(IMO,M,N,A,1,1,IM,B)
    3 IF(DABS(R(MNM*2)).LT.UNDFLO) GOTO 4
*
      IF ((MO.EQ.1).OR.((MO.EQ.-1).AND.(M.GE.N))) INF=1
      DET=0D0
      CALL Lin2dsysccmSolver(MNM,R,M,INF,I,JMO,B,DET)
      IF(IMO.EQ.1) GOTO 18
      GOTO 13

    4 IF(MO.EQ.-1) GOTO 8
      DO 5 I=1,MNM-2
      C(I*2)=R(I*2)*R(I*2)+R(I*2+1)*R(I*2+1)
      C(I*2+1)=R(I*2+2)*R(I*2+1)
    5 CONTINUE
      C(M1*2)=R(M1*2)*R(M1*2)+R(MNM*2-1)*R(MNM*2-1)
      INF=1
      DET=0D0
      CALL Lin3dsysccmSolver(M1,C,M,INF,IM,0,B,P,DET)
      DO 7 L=1,IM
      B((L-1)*M+MNM)=R(MNM*2-1)*B((L-1)*M+M1)
      DO 6 I=M1,2,-1
      B((L-1)*M+I)=R(I*2-1)*B((L-1)*M+I-1)+R(I*2)*B((L-1)*M+I)
    6 CONTINUE
      B((L-1)*M+1)=R(2)*B((L-1)*M+1)
    7 CONTINUE
      GOTO 13
C
    8 DO 9 I=MNM,3,-1
      C(I*2-2)=R(I*2)*R(I*2)+R(I*2-1)*R(I*2-1)
      C(I*2-3)=R(I*2-2)*R(I*2-1)
    9 CONTINUE
      C(2)=R(4)*R(4)+R(3)*R(3)
      DO 10 J=1,IM
      DO 10 I=1,M1
   10 B((J-1)*M+I)=B((J-1)*M+I+1)
      INF=1
      DET=0D0
      CALL Lin3dsysccmSolver(M1,C,M,INF,IM,0,B,P,DET)
      DO 12 L=1,IM
      B((L-1)*M+1)=R(3)*B((L-1)*M+1)
      DO 11 I=2,M1
      B((L-1)*M+I)=R(I*2+1)*B((L-1)*M+I)+R(I*2)*B((L-1)*M+I-1)
   11 CONTINUE
      B((L-1)*M+MNM)=R(2*MNM)*B((L-1)*M+MNM)
   12 CONTINUE
```

```
C
   13 IF(M.EQ.N) GOTO 17
      NI=N*(I-1)
      IF(M.GT.N) THEN
      L=1
      DO 14 K=N,NI,N
      L=L+1
      DO 14 J=1,N
   14 B(K+J)=B((L-1)*M+J)
      ELSE
      L=I+1
      DO 15 K=NI,N,-N
      L=L-1
      DO 15 J=M,1,-1
      B(K+J)=B((L-1)*M+J)
   15 B((L-1)*M+J)=0D0
      DO 16 J=M+1,N
   16 B(NI+J)=0D0
      ENDIF
   17 CALL BTDBK(4,M,N,A,-1,1,IM,B)
      IF(MO.GT.0) GOTO 18
      IF(MSCL(1,MXM,IM,B,EX).EQ.0) CONTINUE
*
   18 INF=0
      RETURN
  100 FORMAT(' M OR N LESS THEN 0 ON PROGRAM PseudsysccmSolver: M=',
     * I2,' N=',I2)
      END
C
C Функция RL - вычисления величину D=|D1+D2+D3|-|Y|
C
C Обращение:   RL(D1,D2,D3,Y)
C Входные данные:
C   D1,D2,D3,Y - (real*8) вещественные данные.
C Выходные данные:
C           RL - (real*8) содержит  величину D.
C
      REAL FUNCTION RL*8(D1,D2,D3,Y)
      REAL*8 D1,D2,D3,D,Y
      IF(DABS(D1).LT.DABS(D2)) GOTO 3
      IF(DABS(D2).LT.DABS(D3)) GOTO 2
    1 D=DABS(D1+D2+D3)
      GOTO 4
    2 D=DABS(D1+D3+D2)
      GOTO 4
    3 IF(DABS(D3).LT.DABS(D1)) GOTO 1
      D=DABS(D2+D3+D1)
    4 IF(Y.GT.1D0) GOTO 5
      RL=DABS(Y-D)
```



```
      RETURN
    5 RL=DABS(1D0-D/Y)
      RETURN
      END
C
C Программы из библиотеки LINA [10] (А.Н.Малышев, 1990).
C                   Программа BTD
C
C Приведение вещественной матрицы к двухдиагональному виду или
C симметричной вещественной матрицы к трехдиагональному виду
C преобразованиями отражения Хаусхолдера.
C
C Обращение:   CALL BTD(I23,M,N,A,B)
C Входные данные:
C          I23 - (integer) равно 2, если приведение к
C                двухдиагональному виду; равно 3, если приведение
C                к трехдиагональному виду;
C            A - (real*8) иходная вещественная матрица;
C            M - (integer) число строк матрицы A;
C            N - (integer) число столбцов матрицы A (при I23=3
C                должно быть N=M).
C Выходные данные:
C            B - (real*8) массив длины 2*L, где L=min(M,N).
C                B(2),B(4),...,B(2*L) - элементы главной диагонали;
C                B(3),B(5),...,B(2*L-1) - элементы побочной диагонали;
C            A - (real*8) в этом массиве упакованы векторы нормалей
C                отражений;
C
      SUBROUTINE BTD(I23,M,N,A,B)
      INTEGER I23,M,N,INC,I,II,I1,L,LL,L1,J,K,K1,K2
      REAL*8 A(1),B(1),RFLD,ZERO
      LOGICAL I2
      DATA ZERO/0D0/
      I2=I23.EQ.2
      B(1)=ZERO
      K2=1
      K=I23-1
      L=M+1-K
      IF(L.LE.0) GOTO 5
      LL=N
      I=1
      II=M
      I1=1
      J=1
      INC=K
      IF(M.GE.N) GOTO 3
    1 I1=II
      II=I
      I=I1
```

```
      IF(I2) GOTO 2
      IF(I.EQ.1) GOTO 2
      B(J-1)=A(K1-1)
      I1=1
      GOTO 4
    2 L1=LL
      LL=L
      L=L1
    3 J=J+INC
      B(J)=RFLD(L,A(K),I)
      K1=K
      K2=K+II
      L1=LL-1
    4 LL=LL-1
      IF(LL.LE.0) GOTO 5
      K=K+II
      IF(L.GT.1) CALL REFL(L,A(K1),I1,L1,A(K2),I,II)
      GOTO 1
    5 IF(.NOT.I2) B(N+N)=A(K2)
      RETURN
      END
C
C                     Программа BTDBK
C
C Применение преобразований отражений Хаусхолдера, вычисленных
C в подпрограмме BTD, к вещественной матрице
C
C Обращение:   CALL BTDBK(I234,M,N,A,I,LR12,K,X)
C Входные данные:
C          A - (real*8) матрица, содержащая векторы нормалей
C               отражений, упакованные подпрограммой BTD;
C          M - (integer) число строк матрицы A;
C          N - (integer) число столбцов матрицы A (при I23=3
C              должно быть N=M).
C       I234 - (integer) равно 2, если применяются левые отражения
C              из процесса двухдиагонализации; равно 4, если
C              применяются правые отражения из процесса
C              двухдиагонализации; равно 3, если применяются левые
C              отражения из процесса трехдиагонализации;
C          I - (integer) равно 1, если отражения применяются в прямом
C              порядке; равно -1, если отражения применяются в
C              обратном порядке;
C          X - (real*8) исходная матрица, к которой применяются
C              преобразования отражения, упакованные в матрице A;
C       LR12 - (integer) равно 1, если отражения применяются к
C              к матрице X слева; равно 2, если отражения
C              применяются к матрице X справа;
C          K - (integer) при LR12=1 число столбцов, а при LR12=2
C              число строк матрицы X.
```



```
C Выходные данные:
C            X - (real*8) содержит результат применения преобразований
C                отражений к исходной матрице X.
C
      SUBROUTINE BTDBK(I234,M,N,A,I,LR12,K,X)
      INTEGER I234,M,N,I,LR12,K
      REAL*8 A(1),X(1)
      INTEGER IA,IX,JX,L,LL,LLL,J,JJ,MP1I,IXI
      LOGICAL I4
      I4=I234.EQ.4
      IA=1
      IF(I4) IA=M
      IX=K
      JX=1
      IF(LR12.EQ.2) GO TO 1
      IX=1
      JX=M
      IF(I4) JX=N
    1 L=MAX0(M,N)
      LL=MIN0(M,N)
      J=1
      JJ=1
      IF((I234.EQ.2).AND.(M.GE.N).OR.(I4.AND.(M.LT.N))) GOTO 2
      L=LL-1
      LL=L-1
      J=J+IX
      JJ=JJ+IA
      GOTO 3
    2 IF(M.EQ.N) LL=LL-1
    3 IF(LL.LE.0) GOTO 6
      MP1I=(M+1)*I
      IXI=IX*I
      IF(I.EQ.1) GOTO 4
      LLL=LL-1
      L=L-LLL
      J=J-IXI*LLL
      JJ=JJ-MP1I*LLL
    4 DO 5 LLL=1,LL
      CALL REFL(L,A(JJ),IA,K,X(J),IX,JX)
      L=L-I
      J=J+IXI
    5 JJ=JJ+MP1I
    6 RETURN
      END
C
C                     Программа REFL
C
C Применение одного преобразования отражения Хаусхолдера к
C вещественной матрице
```

```
C
C Обращение:   CALL REFL(L,P,IP,N,X,IX,JX)
C Входные данные:
C            L - (integer) число компонент в нормали, определяющей
C                преобразование отражения;
C            P - (real*8) массив, содержащий компоненты P(1),P(1+IP),
C                 ...,P(1+(L-1)*IP) вектора нормали отражения;
C           IP - (integer) интервал между компонентами вектора
C                нормали отражения в массиве P;
C            N - (integer) число столбцов в преобразуемой матрице;
C            X - (real*8) массив, содержащий элементы
C
C       X(1),         X(1+JX),          ...,X(1+(N-1)*JX),
C       X(1+IX),      X(1+IX+JX),       ...,X(1+IX+(N-1)*JX),
C         :             :                   :    :
C       X(1+(L-1)*IX),X(1+(L-1)*IX+JX),...,X(1+(L-1)*IX+(N-1)*JX);
C
C           IX - (integer) интервал внутри столбцов между компонентами
C                преобразуемой матрицы в массиве X;
C           JX - (integer) интервал внутри строк между компонентами
C                преобразуемой матрицы в массиве X.
C Выходные данные:
C            X - (real*8) в элементах X(1+I*IX+J*JX), где I=0,1,...,L-1,
C                J=0,1,...,N-1, содержится преобразованная матрица
C
      SUBROUTINE REFL(L,P,IP,N,X,IX,JX)
      INTEGER I,J,K,IPL,JXN,L,IP,N,IX,JX
      REAL*8 INPR,S,P(*),X(*)
      IPL=IP*L
      JXN=JX*N
      DO 1 J=1,JXN,JX
      S=INPR(L,P,IP,X(J),IX)
      K=J
      DO 1 I=1,IPL,IP
      X(K)=X(K)-S*P(I)
    1 K=K+IX
      RETURN
      END
C
C                    Функция RFLD
C
C Построение преобразования отражения Хаусхолдера, которое заданный
C вещественной вектор переводит в вектор, имеющий нулевые компоненты
C со второй по последнюю включительно
C
C Обращение:   RFLD(L,P,IP)
C Входные данные:
C            L - (integer) размерность заданного вектора;
C            P - (real*8) массив, содержащий компоненты P(1),P(1+IP),
```



```
C                ...,P(1+(L-1)*IP) заданного вектора, по которому
C                строится преобразование отражения;
C           IP - (integer) интервал между компонентами заданного
C                вектора в массиве P.
C Выходные данные:
C            P - (real*8) в элементах P(1),P(1+IP),...,P(1+(L-1)*IP)
C                получены компоненты вектора нормали, определяющего
C                преобразования отражения;
C         RFLD - (real*8) первая компонента вектора, полученного при
C                отражении заданного вектора
C
      REAL FUNCTION RFLD*8(L,P,IP)
      INTEGER I,IPL,L,IP
      REAL*8 PM,T,ZERO,INPR,P(*)
      DATA ZERO/0D0/
      IF(L.EQ.1) GO TO 5
      IPL=IP*L
      PM=ZERO
      DO 1 I=1,IPL,IP
    1 PM=DMAX1(PM,DABS(P(I)))
      IF(PM.LE.ZERO) GO TO 5
      DO 2 I=1,IPL,IP
    2 P(I)=P(I)/PM
      T=DSQRT(INPR(L,P,IP,P,IP))
      IF(P(1).LE.ZERO) T=-T
      RFLD=-T*PM
      P(1)=P(1)+T
      PM=DSQRT(P(1)*T)
      DO 3 I=1,IPL,IP
    3 P(I)=P(I)/PM
    4 RETURN
    5 RFLD=P(1)
      P(1)=ZERO
      GO TO 4
      END
C
C                    Функция MSCL
C
C Вычисления масштабного разложения вещественной матрицы или
C умножение вещественной матрицы на степень основания счисления
C машинных вещественных чисел
C
C Обращение:   MSCL(I01,M,N,A,E)
C Входные данные:
C          I01 - (integer) равно 0, если вычисляется масштабное
C                разложение; равно 1, если выполняется умножение
C                на степень основания счисления;
C            A - (real*8) исходная обрабатываемая матрица;
C            M - (integer) число строк матрицы A;
```

```
C           N - (integer) число столбцов матрицы A;
C           E - (integer) при I01=1 показатель степени основания
C               счисления.
C Выходные данные:
C           A - (real*8) при I01=0 результат масштабирования исходной
C                матрицы A; при I01=1 равно произведению исходной
C                матрицы A на степень основания счисления с
C                показателем E;
C           E - (integer) при I01=0 экспонента в масштабном разложении
C               исходной матрицы A;
C        MSCL - (integer) равно -1, если умножение на степень
C               основания невозможно из-за переполнения (в этом
C               случае матрица A на выходе равна исходной);
C               равно 0, если на входе A=0;
C               равно 1 в остальных случаях.
C
      INTEGER FUNCTION MSCL(I01,M,N,A,E)
      INTEGER MN,I,EE,E,I01,M,N
      REAL*8 AM,ZERO,GTEXP,CHEXP,A(*)
      INTEGER RADIX,MAXEXP,MINEXP,MANTSZ
      COMMON /ICONST/RADIX,MAXEXP,MINEXP,MANTSZ
      DATA ZERO/0D0/
      MN=M*N
      AM=ZERO
      DO 1 I=1,MN
    1 AM=DMAX1(AM,DABS(A(I)))
      IF(AM.LE.ZERO) GO TO 7
      AM=GTEXP(AM,EE)
      IF(I01.NE.0) GO TO 2
      E=EE
      EE=-EE
      GO TO 3
    2 IF(E.GT.MAXEXP-EE) GO TO 8
      EE=E
    3 IF(EE.EQ.0) GO TO 5
      DO 4 I=1,MN
    4 A(I)=CHEXP(A(I),EE)
    5 MSCL=1
    6 RETURN
    7 IF(I01.EQ.0) E=0
      MSCL=0
      GO TO 6
    8 MSCL=-1
      GO TO 6
      END
C
C                    Функция INPR
C
C Вычисления скалярного произведения вещественных векторов
```



```
C
C Обращение:   INPR(L,X,IX,Y,IY)
C Входные данные:
C           L - (integer) число компонент у перемножаемых векторов;
C           X - (real*8) массив, содержащий в элементах X(1),X(1+IX),
C               ...,X(1+(L-1)*IX) компоненты первого вектора;
C          IX - (integer) интервал между компонентами первого
C               вектора в массиве X;
C           Y - (real*8) массив, содержащий в элементах Y(1),Y(1+IY),
C               ...,Y(1+(L-1)*IY) компоненты второго вектора;
C          IY - (integer) интервал между компонентами второго
C               вектора в массиве Y.
C Выходные данные:
C        INPR - (real*8) равно X(1)*Y(1)+X(1+IX)*Y(1+IY)+...
C               +X(1+(L-1)*IX)+Y(1+(L-1)*IY)
C
      REAL FUNCTION INPR*8(L,X,IX,Y,IY)
      INTEGER I,J,IXL,L,IX,IY
      REAL*8 ZERO,X(*),Y(*)
      DATA ZERO/0D0/
      INPR=ZERO
      IXL=IX*L
      J=1
      DO 1 I=1,IXL,IX
      INPR=INPR+X(I)*Y(J)
    1 J=J+IY
      RETURN
      END
```

Рукопись поступила в издательский отдел
23 ноября 2000 года.