НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЕ УЧРЕЖДЕНИЕ
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им. Г.И.Будкера СО РАН

Э.А. Бибердорф, Н.И. Попова

ВЫЧИСЛЕНИЯ С ГАРАНТИРОВАННОЙ ОЦЕНКОЙ ТОЧНОСТИ

Часть вторая

РЕШЕНИЕ СПЕКТРАЛЬНЫХ ЗАДАЧ

ИЯФ 2001-21

НОВОСИБИРСК
2001

ВЫЧИСЛЕНИЯ С ГАРАНТИРОВАННОЙ ОЦЕНКОЙ ТОЧНОСТИ
часть вторая

РЕШЕНИЕ СПЕКТРАЛЬНЫХ ЗАДАЧ

Н.И. Попова
Институт ядерной физики им. Г.И. Будкера, 630090, Новосибирск

Э.А. Бибердорф
Институт Математики СО РАН, Россия

В настоящей работе описаны алгоритмы нового типа для вычисления собственных значений и векторов симметричной матрицы, а также основные принципы их построения. Наиболее важными особенностями этих алгоритмов являются 1) учет и суммирование неизбежно возникающих при счете машинных погрешностей, что позволяет *наряду с результатом привести также оценку его точности*, 2) возможность содержательной диагностики аварийных ситуаций с *указанием в сообщении об аварии ее вероятной причины.*

Мы широко используем результаты и рекомендации законченной теории современных вычислительных методов линейной алгебры, разработанной в Институте Математики СО РАН под руководством акад. РАН С.К.Годунова.

В приложении приводится описание созданного нами на языке ФОРТРАН-90 пакета программ, реализующих эти и другие современные алгоритмы линейной алгебры.

CALCULATIONS WITH THE GUARANTEED ESTIMATE OF ACCURACY
part 2

SOLUTION OF SPECTRAL PROBLEMS

E.A. Biberdorf, N.I. Popova

Abstract

In this work we describe the algorithms of new type for the calculation of eigen values and eigen vectors of a symmetric matrix and the basic principles of their construction. From the most importent features of the new method we'll remark out two: 1) all the computer rounding errors that inevitably appear during calculatioms are taken into account and summed and along with the result an estimate of it's accuracy is carried out; 2) in emergency situation a diagnostick program starts and shows a description on possible cause of fault.

We widely used results and recommendations of completed theory of such algorithms that was created under leadership of academician S.K.Godunov in institute of Mathematics SB RAS.

Developed by us a FORTRAN-90 based program package is described in the appendix. The package realize these and other modern algorithms of linear algebra.

1 Введение

Данная работа продолжает начатую в 1999 году (см. [3]) деятельность по программной реализации алгоритмов линейной алгебры нового поколения и посвящена методам вычисления собственных функций и значений симметричных операторов.

Новые алгоритмы отличаются от общепринятых тем, что результат счета сопровождается гарантированной оценкой его точности. Это свойство делает их незаменимыми при численных исследованиях физических процессов, которые требуют нахождения каких-либо спектральных характеристик. Задачи такого рода решаются, как правило, в несколько этапов: 1) физический или численный эксперимент, из которых определяются исходные данные задачи, 2) интерполяция этих

данных на всю интересующую область изменения параметров, 3) дискретизация, т.е. построение конечномерной математической модели и 4) решение спектральной задачи линейной алгебры:

$$Av = \lambda v. \tag{1}$$

Эту схему мы продемонстрируем на примере и обсудим в разделе 4.1.

Каждый шаг решения неизбежно сопровождается характерными именно для него погрешностями, совокупность которых и определяет погрешность конечного результата.

Отметим, что длительная история становления и развития культуры физического эксперимента (этап 1) привела к тому, что любые данные измерений считаются достоверными только в том случае, если они сопровождаются указанием интервала возможной ошибки. Способы оценок ошибок интерполяции и дискретизации (этапы 2,3) также хорошо разработаны (см., в частности, [5],[4]) и являются традиционной частью курсов вычислительной математики в ВУЗах.

Удивительно, что до недавнего времени вычислительным погрешностям последнего этапа схемы не придавалось никакого значения. Результат решения спектральной задачи (1) принимался за абсолютно точный, если, конечно, свойства искомой спектральной характеристики не противоречили физическому смыслу и ошибка не была очевидной.

Отсутствие оценки точности решения спектральной алгебраической задачи делает, по-существу, бессмысленными усилия по повышению точности физического эксперимента и дискретной математической модели. Очевидно, что только использование алгоритмов с гарантированной оценкой точности может исправить ситуацию и позволит провести полный и грамотный анализ степени соответствия результатов численных исследований и физической реальности.

В ближайшее время новые алгоритмы могут найти свое применение в еще одной важной области, связанной с изучением квантовых явлений. Современный уровень точности физических экспериментов здесь очень высок и математическое моделирование призвано оценить достоверность существующих теорий. В настоящий момент в этом направлении ведется целый ряд исследований и уже были выявлены случаи расхождения результатов вычислений и измерений (мы благодарим А.С.Елховского за относящиеся к этой теме обсуждения). Очевидно, что подобные сравнения представляют научный интерес только тогда, когда результаты не только физического, но и численного экспериментов сопровождаются оценками их точности.

Есть основания утверждать, что знание и использование алгоритмов линейной алгебры с гарантированной оценкой точности результата должно стать неотъемлемой частью математической культуры любого исследователя.

К сожалению, даже многие специалисты в вычислительной линейной алгебре не уделяют должного внимания оценкам точности вычислений. Причина этого явления заключается в традиционном подходе к решению задач вида (1), который сложился задолго до наступления эры компьютеров, когда вопрос о вычислительных погрешностях вообще не ставился. Показательным в этом плане является популярное до сих пор пособие [8]. С распространением ЭВМ, расширением практического круга применения задач вида (1) и ростом размеров матрицы A вычислителям все чаще встречались примеры явно неправильной работы классических алгоритмов, что подталкивало математиков к поиску способов оценки погрешностей вычисления.

Первым существенным прорывом в этой области стало изобретение метода обратного анализа погрешностей. В этом методе погрешности арифметических операций трактуются как возмущения начальных данных (в случае задачи (1) – коэффициентов матрицы A). Чрезвычайно подробно этот подход представлен в монографии [7]. На первый взгляд может показаться, что тем самым вопрос об учете арифметических погрешностей решен. Оказалось, однако, что для подавляющего большинства алгоритмов метод обратного анализа практически неприменим из-за его громоздкости.

И только в начале 80-х годов группа математиков под руководством академика РАН С.К. Годунова в ИМ СО РАН приступила к разработке алгоритмов, отвечающих современным требованиям учета вычислительных погрешностей. Новые методы используют как традиционные так и оригинальные нестандартные идеи и при этом достаточно просты. К сожалению, они до сих пор не получили широкого распространения. Сегодня ситуация меняется и многие специалисты начинают по достоинству оценивать преимущества этих алгоритмов. Об этом свидетельствует, в частности, их несомненное признание на конгрессе III International Workshop on Accurate Solution of Eigenvalue Problems, состоявшимся в июле 2000 г. в городе Хаген, Германия, участником которого был один из авторов (Э.А.Бибердорф).

Цель нашей работы заключается в развитии и популяризации указанных алгоритмов. Мы стремились тщательно проработать все детали, влияющие на накопление и учет погрешностей. Особое внимание уделялось тем важным моментам, которые обычно обходятся стороной в статьях и монографиях. При этом мы старались максимально упростить изложение основных идей, что должно сделать новые алгоритмы доступными для неспециалистов.

Пакет программ GALA (Guaranted Accuracy in Linear Algebra), описанный в работе [3], представляет собой реализацию современных вычислительных алгоритмов линейной алгебры нового поколения, позволяющих оценивать точность вычислений. Пакет написан на языке ФОРТРАН–90. Теперь он дополнен представленным в настоящей работе модулем для решения симметричных спектральных задач. Пакет GALA размещен в директории /usr/local/GALA суперкомпьютера ALPHA Института Ядерной физики СО РАН, и открыт для свободного пользования. К нему прилагается также ряд демонстрационных примеров.

2 Необходимые сведения

Нам будут нужны следующие факты и обозначения.

Линейная алгебра

* Основным объектом рассмотрения является вещественная симметричная матрица $A = A^*$ размера M, где $A^* = \bar{A}^T$ матрица сопряженная к A, $\bar{A}$ – матрица комплексносопряженная к A, A^T – транспонированная матрица; a_{ij} – обозначения для элементов матрицы A; главная диагональ матрицы – a_{ii}, $i = 1, \ldots, M$; побочная диагональ – a_{ii+1}, $i = 1, \ldots, M-1$.

* Спектр симметричной матрицы A вещественный. Для собственных значений примем обозначения $\lambda_j(A)$, $j = 1, \ldots, M$. Считаем, что собственные числа пронумерованы по возрастанию $\lambda_1(A) \leq \lambda_2(A) \leq \ldots \leq \lambda_M(A)$.

* Собственные векторы матрицы A будут обозначаться v_j: $Av_j = \lambda_j v_j$.

* Любая симметричная матрица A ортогональными преобразованиями может быть приведена к трехдиагональному виду $S = PAP^*$. В следующем разделе приводится способ построения такого преобразования, использующий операторы отражения (см. [3] разделы 2.1, 3.2).

* Выражение $Av_j = \lambda_j v_j$ (учитывая, что $PP^* = I$) порождает цепочку равенств

$$Sw_j = SPv_j = PAP^*Pv_j = \lambda_j Pv_j = \lambda_j w_j, \tag{2}$$

  из которой следует, что λ_j также является собственным значением трехдиагональной симметричной матрицы S, а $w_j = Pv_j$ – ее собственный вектор. Это позволяет при вычислении собственных чисел и векторов перейти от произвольной симметричной матрицы к трехдиагональной.

* Евклидовой нормой вектора x называется число $\|x\| = \sqrt{\sum_{i=1}^{M} |x_i|^2}$.

* Операторной нормой матрицы A называется число $\|A\| = \max_{\|x\|=1} \|Ax\|$.

* Эквивалентными нормами матрицы являются следующие ее характеристики:

$$\mathcal{M}(A) = \max \left\{ \max_i \sum_{j=1}^{M} |a_{ij}|, \ \max_j \sum_{i=1}^{M} |a_{ij}| \right\}$$

  – максимум по строкам и столбцам сумм модулей элементов соответствующих строк и столбцов;

$$\mathcal{F}(A) = \sqrt{\sum_{i,j=1}^{M} |a_{ij}|^2}$$

  – фробениусова норма матрицы A.

* Приведенные эквивалентные нормы связаны неравенствами

$$\|A\| \le \mathcal{M}(A) \le \sqrt{M}\|A\|, \qquad \|A\| \le \mathcal{F}(A) \le \sqrt{M}\|A\|.$$

* Если S – трехдиагональная матрица порядка M

$$S = \begin{pmatrix} d_1 & b_2 & & & & 0 \\ b_2 & d_2 & b_3 & & & \\ & b_3 & d_3 & b_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & b_{M-1} & d_{M-1} & b_M \\ 0 & & & & b_M & d_M \end{pmatrix}, \tag{3}$$

то имеет место оценка

$$\|S\| \le \mathcal{M}(S) \le \sqrt{3}\|S\|.$$

* Спектр любой матрицы A лежит внутри круга радиуса $\|A\|$: $|\lambda_j(A)| \le \|A\|$.

* Для симметричной трехдиагональной матрицы (3) можно указать более точные границы спектра: $X(S) \le \lambda_j(S) \le Y(S)$, где

$$\begin{aligned} X(S) &= \min \begin{cases} d_1 - |b_2|, \\ \min_{2\le i\le M-1}\left(d_i - |b_i| - |b_{i+1}|\right), \\ d_M - |b_M|, \end{cases} \\ Y(S) &= \max \begin{cases} d_1 + |b_2|, \\ \max_{2\le i\le M-1}\left(d_i + |b_i| + |b_{i+1}|\right), \\ d_M + |b_M|. \end{cases} \end{aligned} \tag{4}$$

* Анализ погрешностей, возникающих при вычислении собственных значений симметричной трехдиагональной матрицы (подробности см. в [3] раздел 4.6), базируется на следующей теореме:

Теорема 1. *Если A и $\mathcal{E}$ – симметричные $M \times M$- матрицы, то*

$$|\lambda_j(A+\mathcal{E}) - \lambda_j(A)| \le \|\mathcal{E}\|,$$

где $\lambda_j(A+\mathcal{E}), \lambda_j(A)$ $(1 \le j \le M)$ – собственные значения матриц $A+\mathcal{E}$ и A соответсвенно.

Таким образом, если симметричную матрицу A возмутить на $\mathcal{E}$, то спектр возмущенной матрицы $A+\mathcal{E}$ будет отличаться от спектра исходной A не более, чем на $\|\mathcal{E}\|$. Это свойство означает, что спектр симметричных матриц устойчив относительно возмущений элементов матрицы.

Машинная арифметика

При реализации на ЭВМ численного алгоритма неизбежно возникают вычислительные погрешности. Причина этого в том, что вещественное число при размещении в машинной памяти как правило заменяется машинным числом, близким к исходному. Для оценок вычислительных погрешностей будем пользоваться следующими сведениями.

* Любое машинное вещественное число z имеет вид:

$$z = \pm\gamma^{p(z)} m_\gamma(z)\ , \tag{5}$$

где γ – целое положительное число и называется основанием машинной арифметики, целое число $p(z)$ $(p_- \le p(z) \le p_+)$ называется γ-ичным порядком, γ-ичная мантисса $m_\gamma(z)$ представима в виде суммы правильных дробей

$$m_\gamma(z) = \frac{a_1}{\gamma} + \frac{a_2}{\gamma^2} + \ldots + \frac{a_k}{\gamma^k},$$

где a_j – целые числа:

$$1 \le a_1 \le \gamma - 1, \;\; 0 \le a_j \le \gamma - 1 \;\; (2 \le j \le k).$$

(В приведенных формулах p_-, p_+ и k от числа z не зависят.)

* Для оценок машинных погрешностей удобно пользоваться следующими величинами

$$\varepsilon_0 = \gamma^{p_-}\frac{1}{\gamma}, \;\; \varepsilon_\infty = \gamma^{p_+}(1 - \frac{1}{\gamma^k}), \;\; \varepsilon_1 = \gamma^1\frac{1}{\gamma^k} \;\; .$$

Из их определения следует, что ε_0 – минимальное положительное машинное число; ε_∞ – максимальное машинное число; ε_1 – положительное число, минимальное из всех машинных чисел $z_{\text{маш}}$ таких, что

$$(1 + z_{\text{маш}})_{\text{маш}} > 1.$$

* Над машинными вещественными числами можно определить следующие операции: результат операции $Fr(z)$ есть вещественное машинное число равное мантиссе числа z, результатом операции $Ex(z)$ является целое число – порядок числа z. Эти операции позволяют проводить вычисления отдельно для мантисс и порядков. Такой подход позволяет искусственным образом существенно повысить точность вычислений и называется арифметикой вынесенных порядков.

* Бинарные арифметические операции над машинными числами будем обозначать

$$\begin{array}{ll} a \oplus b = (a + b)_{\text{маш}}, & a \ominus b = (a - b)_{\text{маш}}, \\ a \otimes b = (a \times b)_{\text{маш}}, & a \oslash b = (a/b)_{\text{маш}}. \end{array} \tag{6}$$

* Погрешности машинных арифметических операций можно смоделировать следующим образом: если $\upsilon \in \{+, -, \times, \div\}$ – обозначение для одной из бинарных операций, a, b – машинные числа, то

$$(a \;\upsilon\; b)_{\text{маш}} = (a \;\upsilon\; b)(1 + \alpha) + \beta\,, \tag{7}$$

где $|\alpha| \le \varepsilon_1$, $|\beta| \le \varepsilon_0$.
Если модуль результата операции больше чем ε_0, то машинная погрешность моделируется равенством

$$(a \;\upsilon\; b)_{\text{маш}} = (a \;\upsilon\; b)(1 + \alpha).$$

* Для того, чтобы предлагаемые ниже алгоритмы имели гарантированную оценку точности результата, использовались специальные машинные операции. Это в первую очередь операции с направленным округлением, которые удобно обозначать

$$a\underline{\oplus}b, \;\; a\underline{\ominus}b, \;\; a\underline{\otimes}b, \;\; a\underline{\oslash}b, a\overline{\oplus}b, \;\; a\overline{\ominus}b, \;\; a\overline{\otimes}b, \;\; a\overline{\oslash}b$$

Положение черты указывает направление округления: черта сверху означает, что приближенный результат не меньше истинного, а черта снизу – результат должен быть не больше истинного.

* Другой специальной операцией является вычитание без нулевого результата:

$$a \ominus_0 b = \begin{cases} a \ominus b \quad \text{при } a \ominus b \ne 0, \\ \varepsilon_1/\gamma \max\{|a|, |b|\} \quad \text{при } a \ominus b = 0, \end{cases} \tag{8}$$

* Погрешности специальных арифметических операций моделируются следующим образом

$$\begin{array}{ll} a\underline{\ominus}_0 b = a(1 + \alpha_1) - b(1 + \beta_1), & a\overline{\ominus}_0 b = a(1 + \alpha_2) - b(1 + \beta_2), \\ a\underline{\otimes}b = ab(1 + \alpha_3) + \varphi_3, & a\overline{\otimes}b = ab(1 + \alpha_4) + \varphi_4, \\ a\overline{\oslash}b = a/b(1 + \alpha_5) + \varphi_5, & a\underline{\oslash}b = a/b(1 + \alpha_6) + \varphi_6, \end{array} \tag{9}$$

где $|\alpha_j| \le \varepsilon$, $|\beta_k| \le \varepsilon$, $|\varphi_l| \le \varepsilon_0$, $(j = 1, 2, \ldots, 6;\; k = 1, 2;\; l = 3, \ldots, 6)$. Параметр ε зависит от способа реализации операций с округлением и пропорционален ε_1.

3 Алгоритмы

3.1 Приведение симметричной матрицы к трехдиагональному виду с гарантированной оценкой точности результата.

Предварительные замечания

Для того, чтобы вычислить собственные числа и векторы произвольной симметричной матрицы, достаточно уметь решать эту задачу для трехдиагональной матрицы и приводить произвольную симметричную матрицу к трехдиагональному виду (см. (2)). Алгоритм такого приведения использующий преобразования отражения Хаусхолдера (см. [3] разделы 2.1,3.2) кратко излагается ниже.

Пусть $A = A^*$ – исходная симметричная матрица размера M. Алгоритм состоит из M шагов, результатом каждого из которых является матрица $A^{(i)}$. Для того, чтобы получить гарантированную оценку относительной точности результата, в процедуру приведения матрицы к трехдиагональному виду необходимо также включить предварительную нормировку матрицы $A^{(0)} = \rho A$ и разнормировку результата $S = \frac{1}{\rho} A^{(M-2)}$ так, чтобы

$$\frac{4(M-2)o_p(M)}{\Delta_p(M)} \le \mathcal{F}(A^{(0)}),$$

где коэффициенты $\Delta_p(M)$ и $o_p(M)$ определяются по формулам (подробнее см. [3] раздел 4.3)

$$\begin{array}{l}
\delta_1 = \varepsilon_1(M+4)/2\,, \qquad \delta_2 = (1+\varepsilon_1)\delta_1 + \varepsilon_1\,, \\
\delta_3 = \delta_1 + \delta_2 + \delta_1\delta_2\,, \qquad \delta_4 = (1+\delta_2)^2/(1-\delta_3) - 1\,, \\
\delta_5 = \varepsilon_1(1+\delta_2)(1+\delta_4) + \delta_4(1+\delta_2) + \delta_2\,, \\
\delta_6 = (\delta_5\sqrt{2} + \varepsilon_0\sqrt{M})\Big((1+\delta_5)\sqrt{2} + \varepsilon_0\sqrt{M}\Big)\,, \\
\delta_7 = \varepsilon_1(1+\delta_6) + \varepsilon_1(M+2+\varepsilon_1(M+1))(2+\delta_6)\,, \\
\Delta_p(M) = \delta_6 + \delta_7\,, \\
o_p(M) = 2\varepsilon_0\sqrt{M}
\end{array} \tag{10}$$

при условии, что

$$\Delta_p(M) \le 1/4(M-2)^2. \tag{11}$$

Алгоритм трехдиагонализации произвольной симметричной матрицы с оценкой точности результата.

Дано: произвольная симметричная матрица A размера M.

Шаг 0. Нормировка $A^{(0)} = \rho A$.

Шаг 1. По первому столбцу матрицы $A^{(0)}$ построим отражение P_1, которое аннулирует все элементы первого столбца, начиная с третьего, сохраняет первый элемент этого столбца и пересчитывает второй элемент (см. [3] раздел 3.2). Вычислим матрицу $P_1 A^{(0)} P_1$. В произведении $P_1 A^{(0)}$ первая строка остается такой же, как и у матрицы $A^{(0)}$. Поэтому в $A^{(1)} = P_1 A^{(0)} P_1$ все элементы первой строки, начиная с третьего, будут нулевыми. Так как $P_1^* = P_1$, получаем, что матрица $A^{(1)}$ вновь симметрична и имеет следующий вид:

$$A^{(1)} = \begin{pmatrix} d_1 & b_2 & 0 & \dots & 0 \\ b_2 & * & * & \dots & * \\ 0 & * & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & * & * & \dots & * \end{pmatrix},$$

Подматрица, элементы которой обозначны звездочками, – симметричная порядка $M-1$.

Шаг i $(i = 2, M-2)$**.** Подбираем отражение с матрицей P_i, аннулирующее элементы i – го столбца матрицы $A^{(i-1)} = P_{i-1} \dots P_1 A^{(0)} P_1 \dots P_{i-1}$ с номерами от $(i+2)$ – го до M – го, сохраняющее элементы i – го столбца с первого по i – й, пересчитывая при этом $i+1$ – ый элемент. Полагаем $A^{(i)} = P_i A^{(i-1)} P_i = P_i \dots P_1 A^{(0)} P_1 \dots P_i$

Шаг $M-1$**.** Разнормировка: если обозначить $P = P_1 P_2 \dots P_{M-2}$, то результат можно записать в виде трехдиагональной матрицы.

$$S = \frac{1}{\rho} P A^{(0)} P^* = \begin{pmatrix} d_1 & b_2 & & & 0 \\ b_2 & d_2 & b_3 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{M-1} & d_{M-1} & b_M \\ 0 & & & b_M & d_M \end{pmatrix}. \tag{12}$$

Шаг M**.** Оценка погрешности трехдиагонализации.
Вычислить ϵ_T – абсолютную погрешность трехдиагонализации матрицы A по формуле (13). Для оценки относительной погрешности $\widetilde{\epsilon}_T$ достаточно абсолютную погрешность ϵ_T разделить на $\|S\|$.

Результатом выполнения алгоритма является трехдиагональная симметричная матрица S размера M к которой приводится произвольная симметричная матрица A того же размера и гарантированная оценка точности процесса трехдиагонализации.

Оценка погрешностей

В ходе выполнения описанного алгоритма на ЭВМ вместо матрицы S будет найдено некоторое ее приближенние $S_{\text{маш}}$, вместо произведения P преобразований отражения Хаусхолдера – его машинная реализация $P_{\text{маш}}$. Анализ погрешностей приведения симметричной матрицы к трехдиагональному виду при помощи преобразований отражения проводится также, как и анализ процесса двухдиагонализации (см. [3] раздел 4.4). Отличие заключается в том, что общее число применяемых преобразований в данном случае равно $2(M-2)$, где M – порядок матрицы. Следовательно имеет место оценка (аналог оценки (42) в [3])

$$\|S_{\text{маш}} - P_{\text{маш}} A P^*_{\text{маш}}\| \le \epsilon_T, \qquad \text{где} \qquad \epsilon_T = M\varepsilon_0 + \sqrt{M}(2M-3)\Delta_p(M)\|S\|, \tag{13}$$

а $\Delta_p(M)$ определяется по формулам (10).

Для вычисления оценки относительной погрешности $\widetilde{\epsilon}_T$ достаточно (13) разделить на $\|S\|$:

$$\widetilde{\epsilon}_T = \frac{M\varepsilon_0}{\|S\|} + \sqrt{M}(2M-3)\Delta_p(M).$$

При счете удобнее оценивать относительную погрешность сверху, используя эквивалентные нормы матрицы S, например

$$\widetilde{\epsilon}_T \le \frac{\sqrt{3}M\varepsilon_0}{\mathcal{M}(S)} + \sqrt{M}(2M-3)\Delta_p(M).$$

3.2 Вычисление собственных значений трехдиагональной симметричной матрицы с гарантированной оценкой точности результата

Предварительные замечания

Алгоритм вычисления собственных значений $\lambda_1 \le \lambda_2 \le \dots \le \lambda_M$ симметричной трехдиагональной матрицы S размера M (см.(12)), основанный на использовании теоремы Штурма (адаптированный к вычислению сингулярных чисел), был подробно описан в работе [3] в разделах 3.3 и 4.6. Там же проведен детальный анализ арифметических погрешностей. Коротко остановимся на его

основных моментах. Напомним, что все собственные значения симметричной трехдиагональной матрицы S вещественны и принадлежат интервалу $[X(S), Y(S)]$ (см. (4)). По теореме Штурма число собственных значений $\lambda_j(S)$ $(1 \le j \le M)$ симметричной трехдиагональной матрицы S, расположенных левее фиксированного числа λ: $\lambda_j(S) < \lambda$, совпадает с числом неположительных элементов в числовой последовательности $\mathcal{P}_1(\lambda)$, $\mathcal{P}_2(\lambda)$, ..., $\mathcal{P}_M(\lambda)$, которая определяется рекуррентно:

$$\begin{array}{l} \mathcal{P}_1 = \frac{|b_2|}{d_1 - \lambda}, \\ \mathcal{P}_j(\lambda) = \frac{|b_{j+1}|}{d_j - \lambda - |b_j| \mathcal{P}_{j-1}(\lambda)} \quad (2 \le j \le M-1), \\ \mathcal{P}_M = \frac{1}{d_M - \lambda - |b_M| \mathcal{P}_{M-1}(\lambda)}. \end{array} \tag{14}$$

Таким образом, если среди значений $\mathcal{P}_j(\lambda)$ оказалось p неположительных, то по теореме Штурма $\lambda_p(S) < \lambda \le \lambda_{p+1}(S)$.

При вычислении последовательности Штурма по формулам (14) на ЭВМ может произойти переполнение разрядной сетки машины. Для предупреждения такого рода аварийных ситуаций рекомендуется предварительно провести нормировку исходной S матрицы: $S_1 = [\rho S]_{\text{маш}}$, $\|S_1\| \le 3$, (см. ниже **Шаг 1**) и возмущение матрицы $S_1 \to S_2$ (**Шаг 2** алгоритма), а в формулах (14) использовать специальные арифметические операции (см. (6),(8)):

$$\begin{array}{l} \mathcal{P}_{1\text{маш}}(\lambda) = |b_2| \oslash (d_1 \ominus_0 \lambda), \\ \mathcal{P}_{j\text{маш}}(\lambda) = |b_{j+1}| \oslash (d_j \ominus_0 \lambda \ominus_0 |b_j| \otimes \mathcal{P}_{j-1\text{маш}}(\lambda)) \qquad (2 \le j < M-1), \\ \mathcal{P}_{M\text{маш}}(\lambda) = 1 \oslash (d_M \ominus_0 \lambda \ominus_0 |b_M| \otimes \mathcal{P}_{M-1\text{маш}}(\lambda)). \end{array} \tag{15}$$

Для моделирования арифметических погрешностей используется метод обратного анализа, позволяющий интерпретировать полученный численный результат, как результат точных вычислений с возмущенными начальными данными. В случае элементарных арифметических операций такая интерпретация демонстрируется равенством (7). Применяя этот метод к формулам (15), получим, что числовая последовательность $\mathcal{P}_{j\text{маш}}(\lambda)$ является точной последовательностью Штурма для некоторой несимметричной трехдиагональной матрицы, спектр которой совпадает со спектром симметричной матрицы S_3 такой, что $\|S_3 - S_2\| \le 12\varepsilon_1$ (подробнее в [3], раздел 4.6). Согласно утверждению теоремы 1 имеет место неравенство $|\lambda_n(S_3) - \lambda_n(S_2)| \le \|S_3 - S_2\| \le 12\varepsilon_1$. Это означает, что последовательности Штурма (14), (15) позволяют численно локализовать спектр матрицы S_2 с точностью $12\varepsilon_1$:

Теорема 2 *Пусть в последовательности $\mathcal{P}_{j\text{маш}}(\lambda)$ $(j = 1, \ldots, M)$, оказалось p неположительных членов, тогда имеют место неравенства*

$$\lambda_p(S_2) < \lambda + 12\varepsilon_1, \qquad \lambda - 12\varepsilon_1 \le \lambda_{p+1}(S_2).$$

(доказательство см. в [2] теорема 2.1, глава 4)

Отметим, что рассматриваемый алгоритм является итерационным, и организован так, что с каждым шагом интервал (x_n, y_n), содержащий искомое собственное значение, уменьшается вдвое (метод бисекций). Очевидно, что критерием для прекращения вычислений должна служить малость этого интервала. Поэтому на первом этапе исполнения алгоритма должно быть задано некоторое ε такое, что точность $y_n - x_n \le \varepsilon$ достаточна для завершения вычислительного процесса. Из предыдущих рассуждений ясно, что ε разумно выбрать порядка $\|S_3 - S_2\|$. Мы будем полагать

$$\varepsilon = 3 \cdot 12\varepsilon_1 = 36\varepsilon_1 \tag{16}$$

С учетом сделанных замечаний алгоритм можно организовать следующим образом.

Алгоритм вычисления собственного значения трехдиагональной симметричной матрицы с оценкой точности результата.

Дано: трехдиагональная симметричная матрица S размера M, n – номер собственного значения матрицы $1 \le n \le M$ (напомним, что собственые значения симметричной матрицы считаются занумерованными по возрастанию).

Шаг 1. Нормировка матрицы.

Умножить матрицу S на число ρ, подобранное так, чтобы

$$\frac{1}{\gamma} \le \max_{i,j}\{|(\rho b_i)_{\text{маш}}|\ ,\ |(\rho d_j)_{\text{маш}}|\} < 1, \quad \text{причем } \rho = \gamma^k, \quad \text{где } k - \text{целое}\ , \tag{17}$$

γ – основание системы счисления ЭВМ. Результат **Шага 1** матрица $S_1 = (\rho S)_{\text{маш}}$.

Шаг 2. Возмущение матрицы.

Те элементы главной и побочной диагоналей матрицы S_1, модуль которых не превосходит ε_1/γ следует заменить на $\pm\varepsilon_1/\gamma$ так, чтобы знак элемента сохранился. В итоге будет получена матрица S_2 с диагональными элементами $\widetilde{d}_j$ и наддиагональными $\widetilde{b}_i$, причем будут выполнены неравенства

$$\varepsilon_1/\gamma \le |\widetilde{b}_i|, |\widetilde{d}_j| \le 1.$$

Шаг 3. Выбор критерия сходимости, присвоение границ исходного интервала.

Положить $\varepsilon = 36\varepsilon_1$ – параметр допустимой погрешности.

Присвоить исходные значения границам интервала, содержащего n – ое собственное значение, согласно формулам (4)

$$x_n = X(S_2), \qquad y_n = Y(S_2).$$

Шаг 4. Проверка малости интервала:

если $y_n - x_n \le \varepsilon$, то перейти к **Шагу 6**;

если $y_n - x_n > \varepsilon$, то перейти к **Шагу 5**.

Шаг 5. Вычисление последовательности Штурма.

Присвоить $\lambda = \frac{1}{2}(x_n + y_n)$. Вычислить последовательность Штурма по формулам (15). Пусть p – число неположительных элементов в последовательности $\mathcal{P}_{1\text{маш}}(\lambda), \mathcal{P}_{2\text{маш}}(\lambda), \ldots, \mathcal{P}_{M\text{маш}}(\lambda)$.

Если $n \le p$, то присвоить $y_n = \lambda$.

Если $p < n$, то присвоить $x_n = \lambda$.

Перейти к **Шагу 4**.

Шаг 6. Разнормировка.

Присвоить $\widetilde{\lambda}_n(S_2) = \frac{1}{2}(x_n + y_n)$ – приближенная величина n – го собственного значения матрицы S_2. Результат его разнормировки $\widetilde{\lambda}_n(S) = \frac{1}{\rho}\widetilde{\lambda}(S_2)$ – есть приближенная величина n – го собственного значения исходной матрицы S.

Шаг 7. Оценка погрешности результата.

Определить $\epsilon_\lambda(S)$ – абсолютную погрешность приближенного вычисления собственного значения по формуле (18). Для оценки относительной погрешности $\widetilde{\epsilon}_\lambda(S)$ достаточно неравенство (18) разделить на $\|S\|$.

Результатом выполнения алгоритма является приближенно вычисленное n – ое собственное значение $\widetilde{\lambda}_n(S)$ симметричной трехдиагональной матрицы S и гарантированная оценка точности его вычисления.

Замечание. Оценка точности приближения $\epsilon_\lambda(S)$ позволяет установить гарантированные границы $\lambda_n^{(-)} = \widetilde{\lambda}_n(S) - \epsilon_\lambda(S)$ и $\lambda_n^{(+)} = \widetilde{\lambda}_n(S) + \epsilon_\lambda(S)$ интервала, в котором находится истинное собственное значение: $\lambda_n^{(-)} < \lambda_n(S) < \lambda_n^{(+)}$.

Оценка погрешностей

Общая погрешность $|\lambda_n(S) - \widetilde{\lambda}_n(S)|$ приближенного вычисления n-го собственного значения симметричной трехдиагональной матрицы S оценивается суммой

$$\frac{1}{\rho}|\rho\lambda_n(S) - \lambda_n(S_1)| + \frac{1}{\rho}|\lambda_n(S_1) - \lambda_n(S_2)| + \frac{1}{\rho}|\lambda_n(S_2) - \widetilde{\lambda}_n(S_2)| + |\frac{1}{\rho}\widetilde{\lambda}_n(S_2) - \widetilde{\lambda}_n(S)|.$$

Обратим внимание, что из условия (17) следует оценка для множителя $1/\rho$:

$$1/\rho \le \gamma \max_{i,j}\{|b_i|\ ,\ |d_j|\} \le \gamma \mathcal{M}(S) \le \gamma\sqrt{3}||S||.$$

Заметим, что ошибка будет минимальной, если в качестве ρ выбрать степень γ.

Рассмотрим погрешность, которая вносится в результат в процессе нормировки и разнормировки:

$$\frac{1}{\rho}|\rho\lambda_n(S) - \lambda_n(S_1)| + |\frac{1}{\rho}\widetilde{\lambda}_n(S_2) - \widetilde{\lambda}_n(S)| \le$$

$$\frac{1}{\rho}||\rho S - S_1|| + |\frac{1}{\rho}\widetilde{\lambda}_n(S_2) - \widetilde{\lambda}_n(S)| \le 3\varepsilon_0 \max\{\frac{1}{\rho}, 1\} \le 3\varepsilon_0 \max\{\gamma\sqrt{3}||S||, 1\}.$$

Вклад возмущения на **Шаге 2** алгоритма в общую погрешность оценивается как

$$\frac{1}{\rho}|\lambda_n(S_1) - \lambda_n(S_2)| \le \frac{1}{\rho}||S_1 - S_2|| \le \varepsilon_1\sqrt{3}||S||.$$

Погрешность применения метода бисекций дается слагаемым

$$\frac{1}{\rho}|\lambda_n(S_2) - \widetilde{\lambda}_n(S_2)| \le \frac{1}{\rho}\frac{y_n - x_n}{2} \le 18\varepsilon_1\ \gamma\sqrt{3}||S||.$$

Суммируя последние три оценки получаем, что общая абсолютная погрешность приближенного вычисления n – го собственного значения $\widetilde{\lambda}_n(S)$ симметричной трехдиагональной матрицы S оценивается величиной $\epsilon_\lambda(S)$:

$$|\lambda_n(S) - \widetilde{\lambda}_n(S)| \le \epsilon_\lambda(S),$$

где

$$\epsilon_\lambda(S) = 3\varepsilon_0 \max\{\gamma\sqrt{3}||S||, 1\} + \varepsilon_1\sqrt{3}||S|| + 18\varepsilon_1\ \gamma\sqrt{3}||S|| = \tag{18}$$

$$= 3\varepsilon_0 \max\{\gamma\sqrt{3}||S||, 1\} + \varepsilon_1\sqrt{3}||S||(18\gamma + 1).$$

Для вычисления оценки относительной погрешности

$$\frac{|\lambda_n(S) - \widetilde{\lambda}_n(S)|}{||S||} \le \widetilde{\epsilon}_\lambda(S),$$

достаточно $\epsilon_\lambda(S)$ разделить на $||S||$:

$$\widetilde{\epsilon}_\lambda(S) = 3\varepsilon_0 \max\{\gamma\sqrt{3}, \frac{1}{||S||}\} + \varepsilon_1\sqrt{3}(18\gamma + 1). \tag{19}$$

При счете часто удобнее оценивать относительную погрешность сверху, используя эквивалентные нормы матрицы S, например

$$\widetilde{\epsilon}_\lambda(S) \le 3\varepsilon_0 \max\{\gamma\sqrt{3}, \frac{\sqrt{3}}{\mathcal{M}(\mathcal{S})}\} + \varepsilon_1\sqrt{3}(18\gamma + 1).$$

3.3 Двусторонние последовательности Штурма для трехдиагональных симметричных матриц

Односторонние последовательности Штурма использовались выше для расчета собственных значений трехдиагональной матрицы. Ниже будет показано, что для вычисления собственых векторов необходимы так называемые двусторонние последовательности Штурма.

Пусть дана симметричная трехдиагональная матрица порядка M

$$S = \begin{pmatrix} d_1 & b_2 & & 0 \\ b_2 & d_2 & \ddots & \\ & \ddots & \ddots & b_M \\ 0 & & b_M & d_M \end{pmatrix}. \tag{20}$$

Равенства (14) эквивалентны системе уравнений

$$\begin{gathered}\mathcal{P}_0(\lambda) - (d_1 - \lambda) + |b_2|/\mathcal{P}_1(\lambda) = 0,\\ |b_2|\mathcal{P}_1(\lambda) - (d_2 - \lambda) + |b_3|/\mathcal{P}_2(\lambda) = 0,\\ \dots\\ |b_{M-1}|\mathcal{P}_{M-2}(\lambda) - (d_{M-1} - \lambda) + |b_M|/\mathcal{P}_{M-1}(\lambda) = 0,\\ |b_M|\mathcal{P}_{M-1}(\lambda) - (d_M - \lambda) + 1/\mathcal{P}_M(\lambda) = 0.\end{gathered} \tag{21}$$

Определение 1. *Решение системы (21), удовлетворяющее левому краевому условию $\mathcal{P}_0(\lambda) = 0$, называется левосторонней последовательностью Штурма для трехдиагональной симметричной матрицы (20) и обозначается $\mathcal{P}_0^{(+)}(\lambda), \mathcal{P}_1^{(+)}(\lambda), \dots, \mathcal{P}_M^{(+)}(\lambda)$:*

$$\begin{aligned}&\mathcal{P}_0^{(+)}(\lambda) = 0,\\ &\mathcal{P}_j^{(+)}(\lambda) = \frac{|b_{j+1}|}{d_j - \lambda - |b_j|\mathcal{P}_{j-1}^{(+)}(\lambda)} \quad (1 \le j \le M-1),\\ &\mathcal{P}_M^{(+)}(\lambda) = \frac{1}{d_M - \lambda - |b_M|\mathcal{P}_{M-1}^{(+)}(\lambda)}.\end{aligned} \tag{22}$$

Определение 2. *Правосторонней последовательностью Штурма для матрицы (20) называют решение $\mathcal{P}_0^{(-)}(\lambda), \mathcal{P}_1^{(-)}(\lambda), \dots, \mathcal{P}_M^{(-)}(\lambda)$ системы (21), удовлетворяющее правому краевому условию $\mathcal{P}_M^{(-)}(\lambda) = +\infty$:*

$$\begin{aligned}&\mathcal{P}_M^{(-)}(\lambda) = +\infty,\\ &\mathcal{P}_j^{(-)}(\lambda) = \frac{d_{j+1} - \lambda - |b_{j+2}|/\mathcal{P}_{j+1}^{(-)}(\lambda)}{|b_{j+1}|} \quad (M-1 \ge j \ge 1),\\ &\mathcal{P}_0^{(-)}(\lambda) = d_1 - \lambda - |b_2|/\mathcal{P}_1^{(-)}(\lambda)\ .\end{aligned} \tag{23}$$

Напомним, что элементы левосторонней последовательности Штурма пропорциональны отношениям последовательных главных миноров матрицы $S - \lambda I$: $\mathcal{P}_j^{(+)}(\lambda) = |b_{j+1}|\mathcal{D}_{j-1}(\lambda)/\mathcal{D}_j(\lambda)$, где $\mathcal{D}_j(\lambda)$ – главный минор j – го порядка. Отсюда следует, что левосторонняя последовательность $\mathcal{P}_0^{(+)}(\lambda), \mathcal{P}_1^{(+)}(\lambda), \dots, \mathcal{P}_M^{(+)}(\lambda)$ является также правосторонней последовательностью Штурма, если $\lambda = \lambda_n(S)$ – собственное значение матрицы S. С другой стороны, если $\mathcal{P}_M^{(+)}(\lambda) = +\infty$, то λ – корень характеристического многочлена $\mathcal{D}_M(\lambda)$ матрицы S.

Определение 3. *Последовательность $\mathcal{P}_0(\lambda), \mathcal{P}_1(\lambda), \dots, \mathcal{P}_M(\lambda)$, удовлетворяющая одновременно двум краевым условиям $\mathcal{P}_0(\lambda) = 0$ и $\mathcal{P}_M(\lambda) = +\infty$, называется двусторонней последовательностью Штурма симметричной трехдиагональной матрицы S.*

Замечание. Ясно, что такие последовательности могут быть построены, если только λ является собственным значением матрицы S.

Двусторонние последовательности Штурма симметричной трехдиагональной матрицы могут быть использованы для вычисления компонент собственных векторов матриц. Действительно, система (21) для двусторонней последовательности выглядит следующим образом:

$$\begin{gathered}-(d_1 - \lambda) + |b_2|/\mathcal{P}_1(\lambda) = 0\ ,\\ |b_2|\mathcal{P}_1(\lambda) - (d_2 - \lambda) + |b_3|/\mathcal{P}_2(\lambda) = 0\ ,\\ \dots\\ |b_{M-1}|\mathcal{P}_{M-2}(\lambda) - (d_{M-1} - \lambda) + |b_M|/\mathcal{P}_{M-1}(\lambda) = 0\ ,\\ |b_M|\mathcal{P}_{M-1}(\lambda) - (d_M - \lambda) = 0\ .\end{gathered} \tag{24}$$

В то же время, если $\lambda = \lambda_n$ – собственное значение матрицы S и $v = (v_1, v_2, \dots, v_M)^T$ – соответствующий собственный вектор, то выполнено соотношение $Sv = \lambda v$, покомпонентная запись которого:

$$\begin{gathered}-(d_1 - \lambda) - b_2 v_2/v_1 = 0,\\ -b_{M-j}v_{M-j-1}/v_{M-j} - (d_{M-j} - \lambda) - b_{M-j+1}v_{M-j+1}/v_{M-j} = 0 \quad (1 \le j \le M-2),\\ -b_M v_{M-1}/v_M - (d_M - \lambda) = 0.\end{gathered} \tag{25}$$

Эта система равенств совпадает с (24), если положить

$$\mathcal{P}_j(\lambda_n) = \frac{-sign(b_{j+1})v_j}{v_{j+1}} \quad (1 \le j \le M-1)\ . \tag{26}$$

Таким образом, для определения отношений компонент собственного вектора $v = (v_1, v_2, \ldots, v_M)^T$ соответствующего определенному $\lambda = \lambda_n$ собственному значению матрицы S, достаточно вычислить левосторонюю последовательность Штурма матрицы S, которая одновременно является двусторонней.

3.4 Вычисление двусторонней последовательности Штурма трехдиагональной симметричной матрицы с гарантированной оценкой точности результата

Предварительные замечания

Описанный выше простой способ нахождения отношений компонент собственных векторов не может быть реализован в условиях машинных вычислений из-за неизбежных арифметических погрешностей (см. пример в [2] стр.254). Приближенно вычисленная для собственного значения $\lambda_n(S)$ матрицы S размера M (см.(20)) левосторонняя последовательность Штурма $\mathcal{P}_0^{(+)}(\lambda_n)$, $\mathcal{P}_1^{(+)}(\lambda_n)$, ..., $\mathcal{P}_M^{(+)}(\lambda_n)$ может сильно отличаться от двусторонней. Основная идея необходимой модификации алгоритма заключается в том, что одновременно строятся и право- и левосторонняя последовательности, а затем производится их "склейка". Полученная "склеенная"двусторонняя последовательность является точной двусторонней последовательностью для некоторой матрицы близкой к исходной.

Для того, чтобы во время промежуточных вычислений не произошло аварийных ситуаций, как и при вычислении собственных значений, матрицу S необходимо предварительно нормировать ($S \to S_1$) и возмутить ($S_1 \to S_2$).

Пусть известно, что $\lambda_n^{(-)}$ и $\lambda_n^{(+)}$ – гарантированные нижняя и верхняя границы n-го собственного значения $\lambda_n(S_2)$ матрицы S_2: $\lambda_n^{(-)} \le \lambda_n(S_2) \le \lambda_n^{(+)}$. Опишем формально формулы для вычисления машинных вариантов односторонних последовательностей. В них учтены соответствующие краевые условия для левосторонней и правосторонней последовательностей, и тем самым исключены элементы $\mathcal{P}_0^{(+)}$ и $\mathcal{P}_M^{(-)}$. И в дальнейшем они не будут использоваться в алгоритмах, так как для вычисления отношений компонент собственного вектора необходимы только элементы с номерами $1, \ldots, M-1$ (см. (26)).

Для элементов левосторонней последовательности имеют место формулы:

$$\begin{gathered}
\mathcal{P}_{1\text{маш}}^{(+)} = |\widetilde{b}_2|\overline{\oslash}(\widetilde{d}_1\underline{\ominus}_0\lambda_n^{(+)}), \\
\mathcal{P}_{j\text{маш}}^{(+)} = |\widetilde{b}_{j+1}|\overline{\oslash}(\widetilde{d}_j\underline{\ominus}_0\lambda_n^{(+)}\underline{\ominus}_0|\widetilde{b}_j|\overline{\otimes}\mathcal{P}_{j-1\text{маш}}^{(+)}) \quad (2 \le j \le M-1), \\
\mathcal{P}_{M\text{маш}}^{(+)} = 1\overline{\oslash}(\widetilde{d}_M\underline{\ominus}_0\lambda_n^{(+)}\underline{\ominus}_0|\widetilde{b}_M|\overline{\otimes}\mathcal{P}_{M-1\text{маш}}^{(+)}).
\end{gathered} \tag{27}$$

Аналогичные формулы для элементов правосторонней последовательности:

$$\begin{gathered}
\mathcal{P}_{M-1\text{маш}}^{(-)} = (\widetilde{d}_M\overline{\ominus}_0\lambda_n^{(-)})\overline{\oslash}|\widetilde{b}_M|, \\
\mathcal{P}_{j\text{маш}}^{(-)} = (\widetilde{d}_{j+1}\overline{\ominus}_0\lambda_n^{(-)}\overline{\ominus}_0|\widetilde{b}_{j+2}|\underline{\oslash}\mathcal{P}_{j+1\text{маш}}^{(-)})\overline{\oslash}|\widetilde{b}_{j+1}| \quad (j = M-2, M-3, \ldots, 1), \\
\mathcal{P}_{0\text{маш}}^{(-)} = \widetilde{d}_1\overline{\ominus}_0\lambda_n^{(-)}\overline{\ominus}_0|\widetilde{b}_2|\underline{\oslash}\mathcal{P}_{1\text{маш}}^{(-)}.
\end{gathered} \tag{28}$$

Здесь $\widetilde{d}_j$ – диагональные, а $\widetilde{b}_j$ – наддиагональные элементы возмущенной матрицы S_2, а через $\oslash, \ominus, \otimes$ обозначены соответствующие машинные операции (см. раздел 2). В формулах также присутствуют знаки специальных машинных операций. Их использование необходимо для гарантированной оценки точности вычисления последовательности Штурма и компонент собственного вектора.

Введем вспомогательные последовательности.

Определение 4. *Если* $p_j^{(+)}$ *есть число неположительных элементов среди*

$$\mathcal{P}_1^{(+)},\ \mathcal{P}_2^{(+)},\ \ldots, \mathcal{P}_j^{(+)},$$

то последовательность

$$\varphi_j^{(+)} = p_j^{(+)}\,\pi + \arctan \mathcal{P}_j^{(+)} \tag{29}$$

называется левосторонней последовательностью Штурма второго рода.

Аналогично определяется правосторонняя последовательность Штурма второго рода:

Определение 5. *Пусть* q_j *есть число неположительных элементов среди*

$$\mathcal{P}_{j+1}^{(-)},\ \mathcal{P}_{j+2}^{(-)},\ \ldots, \mathcal{P}_{M-1}^{(-)}$$

и $p_j^{(-)} = n - 1 - q_j$*, тогда последовательность*

$$\varphi_j^{(-)} = p_j^{(-)}\,\pi + \arctan \mathcal{P}_j^{(-)} \tag{30}$$

называется правосторонней последовательностью Штурма второго рода.

Важное свойство последовательностей второго рода – монотонность. Благодаря этому графики последовательностей $\varphi_j^{(+)}$ и $\varphi_j^{(-)}$ пересекаются. Пересечение означает, что для некоторого J имеет место неравенство

$$\varphi_{J-1}^{(+)} \le \varphi_{J-1}^{(-)}.$$

Составим последовательность из тех отрезков односторонних последовательностей, которые удовлетворяют краевым условиям и соединяются в точке пересечения графиков последовательностей второго рода:

$$\mathcal{P}_1^{(+)}, \mathcal{P}_2^{(+)}, \ldots, \mathcal{P}_{J-1}^{(+)}, \mathcal{P}_J^{(-)}, \mathcal{P}_{J+1}^{(-)}, \ldots, \mathcal{P}_{M-1}^{(-)}.$$

Ее можно считать приближением точной двусторонней последовательности.

Алгоритм вычисления двусторонней последовательности Штурма трехдиагональной симметричной матрицы для n – го собственного значения с оценкой точности результата.

Дано: трехдиагональная симметричная матрица S размера M, n – номер собственного значения матрицы $1 \le n \le M$ (собственые значения симметричной матрицы считаются занумерованными по возрастанию).

Шаг 1. Номировка матрицы.
$S_1 = (\rho S)_{\text{маш}}$, где ρ выбирается из условия (17).

Шаг 2. Возмущение матрицы.
Те элементы главной и побочной диагоналей матрицы S_1, модуль которых не превосходит ε_1/γ следует заменить на $\pm\varepsilon_1/\gamma$ так, чтобы знак элемента сохранился. В итоге будет получена матрица S_2 с диагональными элементами $\widetilde{d}_j$ и наддиагональными $\widetilde{b}_i$, причем будут выполнены неравенства

$$\varepsilon_1/\gamma \le |\widetilde{b}_i|, |\widetilde{d}_j| \le 1.$$

Шаг 3. Границы собственного значения.
Определить гарантированные границы $\lambda_n^{(-)}, \lambda_n^{(+)}$ собственного значения $\lambda_n(S_2)$:
$\lambda_n^{(-)} < \lambda_n(S_2) < \lambda_n^{(+)}$, например при помощи алгоритма бисекций (см. раздел 3.2).

Шаг 4. Вычисление последовательностей Штурма.
Вычислить лево – и правоостороннюю последовательности Штурма $\mathcal{P}_{j\text{маш}}^{(+)}$, $\mathcal{P}_{j\text{маш}}^{(-)}$ в граничных точках интервала $(\lambda_n^{(-)}, \lambda_n^{(+)})$ по формулам (27) и (28).

Шаг 5. "Склейка". Вычисление двусторонней последовательности Штурма.

Вычислить последовательности Штурма второго рода:

$$\varphi_j^{(+)} = p_j^{(+)}\,\pi + \arctan \mathcal{P}_{j\text{маш}}^{(+)} \qquad \varphi_j^{(-)} = p_j^{(-)}\,\pi + \arctan \mathcal{P}_{j\text{маш}}^{(-)}.$$

Определить максимальное целое число J такое, что

$$\varphi_{J-1}^{(+)} \le \varphi_{J-1}^{(-)}.$$

Составить последовательность

$$\mathcal{P}_{1\text{маш}}^{(+)},\ \mathcal{P}_{2\text{маш}}^{(+)},\ \ldots,\ \mathcal{P}_{J-1\text{маш}}^{(+)},\ \mathcal{P}_{J\text{маш}}^{(-)},\ \mathcal{P}_{J+1\text{маш}}^{(-)},\ \ldots,\ \mathcal{P}_{M-1\text{маш}}^{(-)}. \tag{31}$$

Шаг 6. Оценка погрешности результата.

Определить $\widetilde{\epsilon}_S$ – относительную погрешность вычисления двусторонней последовательности Штурма матрицы S согласно формуле (38).

Результатом выполнения алгоритма является последовательность (31) – двусторонняя последовательность Штурма матрицы S_2, она же и для исходной матрицы S (см. Замечание) и гарантированная оценка точности ее вычисления.

Замечание. В данном алгоритме отсутствует разнормировка. Это обьясняется тем, что собственные векторы пропорциональных матриц равны. Следовательно элементы двусторонней последовательности Штурма исходной и нормированной матриц совпадают, так как согласно равенству (26) они представляют собой отношения компонент вектора, являющегося собственным для обеих матриц.

Оценка погрешностей

Для того чтобы склеенную последовательность можно было считать двусторонней, необходимо проверить, что возмущение элементов матрицы не слишком велико и имеет относительный характер. Для анализа погрешности вычисления двусторонней последовательности, сначала предположим, что элементы исходной матрицы S удовлетворяют условиям

$$\frac{1}{\gamma} \le \max_{i,j}\{|b_i|\ ,\ |d_j|\} < 1\ ,\ \text{и}\ \varepsilon_1/\gamma \le |b_i|, |d_j|, \tag{32}$$

что делает излишними **Шаги 1** и **2** алгоритма. Другими словами, будем считать, что $S = S_2$ и $\rho = 1$. В этом случае погрешности арифметических операций в формулах (27), (28) можно интерпретировать как возмущения, внесенные в элементы исходной матрицы S. Действительно, учитывая (9), система (27) моделируется следующим образом

$$\begin{aligned}
&|b_j|\overline{\otimes}\mathcal{P}_{j-1\text{маш}}^{(+)} = |b_j|\mathcal{P}_{j-1\text{маш}}^{(+)}(1+\varphi_j) + \xi_j\ ,\\
&d_j\underline{\ominus_0}\lambda_n^{(+)} = (1+\psi_j)d_j - (1+\overline{\psi}_j)\lambda_n^{(+)}\ ,\\
&(d_j\underline{\ominus_0}\lambda_n^{(+)})\underline{\ominus_0}(|b_j|\overline{\otimes}\mathcal{P}_{j-1\text{маш}}^{(+)}) = (1+\chi_j)(d_j\underline{\ominus_0}\lambda_n^{(+)}) - (1+\overline{\chi}_j)(|b_j|\overline{\otimes}\mathcal{P}_{j-1\text{маш}}^{(+)})\ ,\\
&\mathcal{P}_{j\text{маш}}^{(+)} = \frac{|b_{j+1}|(1+\zeta_{j+1})}{d_j\underline{\ominus_0}\lambda_n^{(+)}\underline{\ominus_0}|b_j|\overline{\otimes}\mathcal{P}_{j-1\text{маш}}^{(+)}},
\end{aligned}$$

где $|\varphi_j| \le \varepsilon_1,\ |\xi_j| \le \varepsilon_0, |\psi_j| \le \varepsilon_1,\ |\overline{\psi}_j| \le \varepsilon_1, |\chi_j| \le \varepsilon_1,\ |\overline{\chi}_j| \le \varepsilon_1, |\zeta_{j+1}| \le \varepsilon_1$.

Определим следующие величины

$$\begin{aligned}
&c_2^{(+)} = b_2(1+\zeta_2),\ \ d_1^{(+)} = (1+\psi_1)d_1 - \overline{\psi}_1\lambda_n^{(+)},\\
&c_{j+1}^{(+)} = b_{j+1}(1+\zeta_{j+1})/(1+\overline{\chi}_j),\ \ b_j^{(+)} = (1+\varphi_j)b_j,\ \ d_j^{(+)} = \tfrac{(1+\chi_j)(1+\psi_j)}{(1+\overline{\chi}_j)} - \left(\tfrac{(1+\chi_j)(1+\overline{\psi}_j)}{1+\overline{\chi}_j} - 1\right)\lambda_n^{(+)},
\end{aligned} \tag{33}$$

для них верны оценки

$$|c_{j+1}^{(+)} - b_{j+1}| \le \tfrac{2\varepsilon_1}{1-2\varepsilon_1}|b_{j+1}|\ ,\ \ |b_j^{(+)} - b_j| \le \varepsilon_1|b_j|\ ,\ \ |d_j^{(+)} - d_j| \le \tfrac{3\varepsilon_1}{1-3\varepsilon_1}(|d_j| + |\lambda_n^{(+)}|) + \varepsilon_0\ .$$

Аналогичное моделирование погрешностей в (28) приводит к следующей интерпретации.

Теорема 3. *Вычисленные по формулам (27), (28) элементы последовательностей связаны соотношениями*

$$\mathcal{P}^{(+)}_{1\text{маш}} = |c^{(+)}_2|/(d^{(+)}_1 - \lambda') \ ,$$
$$\mathcal{P}^{(+)}_{i+1\text{маш}} = |c^{(+)}_{i+2}|/(d^{(+)}_{i+1} - \lambda' - |b^{(+)}_{i+1}|\mathcal{P}^{(+)}_{i\text{маш}}) \quad (1 \le i \le M-1),$$
$$\mathcal{P}^{(-)}_{M-1\text{маш}} = (d^{(-)}_M - \lambda')/|b^{(-)}_M| \ ,$$
$$\mathcal{P}^{(-)}_{i\text{маш}} = (d^{(-)}_{i+1} - \lambda') - |c^{(-)}_{i+2}|/\mathcal{P}^{(-)}_{i+1\text{маш}})/|b^{(-)}_{i+1}| \quad (0 \le i \le M-2),$$

в которых $\lambda' = (\lambda^{(-)}_n + \lambda^{(+)}_n)/2$. *Имеют место оценки*

$$\begin{gathered}\frac{|c^{(+)}_i - b_i|}{|b_i|} \le \varepsilon', \quad \frac{|b^{(+)}_i - b_i|}{|b_i|} \le \varepsilon', \quad \frac{|c^{(-)}_i - b_i|}{|b_i|} \le \varepsilon', \\ \frac{|b^{(-)}_i - b_i|}{|b_i|} \le \varepsilon', \quad |d^{(+)}_i - d_i| \le \varepsilon''||S||, \quad |d^{(-)}_i - d_i| \le \varepsilon''||S||,\end{gathered} \tag{34}$$

где

$$\varepsilon' = 2\varepsilon_1, \qquad \varepsilon'' = 3\varepsilon_1. \tag{35}$$

Неравенства (34) означают, что последовательности, построенные по формулам (27), (28), являются точными односторонними последовательностями, но не исходной симметричной матрицы, а двух различных несимметричных трехдиагональных матриц, но близких по норме к исходной.

Склейка этих двух последовательностей производится на основании следующей леммы.

Лемма 1. *Пусть* J $(1 \le J \le M-1)$ *- место пересечения односторонних последовательностей второго рода*

$$\varphi^{(+)}_{J-1} \le \varphi^{(-)}_{J-1},$$

а в составной последовательности

$$\mathcal{P}^{(+)}_{1\text{маш}}, \ \mathcal{P}^{(+)}_{2\text{маш}}, \ \ldots, \ \mathcal{P}^{(+)}_{J-1\text{маш}}, \ \mathcal{P}^{(-)}_{J\text{маш}}, \ \mathcal{P}^{(-)}_{J+1\text{маш}}, \ \ldots, \ \mathcal{P}^{(-)}_{M-1\text{маш}}$$

в точности $n-1$ *неположительных элементов. Тогда существует матрица* $\widetilde{S}$, *для которой последовательность*

$$0 = \mathcal{P}^{(+)}_{0\text{маш}}, \ \mathcal{P}^{(+)}_{1\text{маш}}, \ \mathcal{P}^{(+)}_{2\text{маш}}, \ \ldots, \ \mathcal{P}^{(+)}_{J-1\text{маш}}, \ \mathcal{P}^{(-)}_{J\text{маш}}, \ \mathcal{P}^{(-)}_{J+1\text{маш}}, \ \ldots, \ \mathcal{P}^{(-)}_{M-1\text{маш}}\mathcal{P}^{(-)}_{M\text{маш}} = +\infty$$

является двусторонней последовательностью Штурма:

$$\widetilde{S} = \begin{pmatrix} d^{(+)}_1 & c^{(+)}_2 & & & & & 0 \\ b^{(+)}_2 & d^{(+)}_2 & c^{(+)}_3 & & & & \\ \ddots & \ddots & \ddots & & & & \\ & b^{(+)}_{j-1} & d^{(+)}_{j-1} & c^{(+)}_j & & & \\ & & \widetilde{b}_j & \widetilde{d}_j & \widetilde{c}_{j+1} & & \\ & & & b^{(-)}_{j+1} & d^{(-)}_{j+1} & c^{(-)}_{j+2} & \\ & & & \ddots & \ddots & \ddots & \\ & & & & b^{(-)}_{M-1} & d^{(-)}_{M-1} & c^{(-)}_M \\ 0 & & & & & b^{(-)}_M & d^{(-)}_M \end{pmatrix},$$

причем близость матриц S *и* $\widetilde{S}$ *может быть оценена следующим образом*

$$||S - \widetilde{S}|| \le \mathcal{M}(S - \widetilde{S}) \le 2(\varepsilon' + \varepsilon'')\mathcal{M}(S) + \frac{\lambda^{(+)}_n - \lambda^{(-)}_n}{2}$$

Доказательство этой леммы можно найти в [2] (глава 4). Заметим только, что "склейка"становиться возможной именно за счет вычислительных погрешностей, так как благодаря направленным округлениям результатов арифметических операций в формулах (27) и (28) вычисленные правосторонняя и левосторонняя последовательности Штурма пересекаются.

Из леммы следует, что вычисленную двустороннюю последовательность Штурма матрицы S_2 (**Шаги 4** и **5** алгоритма) можно считать точной двусторонней последовательностью для матрицы $\widetilde{S}_2$, причем с учетом условий (32) и равенств (16) и (35) верна оценка:

$$\|S_2 - \widetilde{S}_2\| \leq 30\varepsilon_1 + 18\varepsilon_1 = 48\varepsilon_1. \tag{36}$$

Погрешности нормировки и возмущения матрицы S оцениваются также как и в разделе 3.2:

$$\frac{1}{\rho}\|\rho S - S_1\| \leq \varepsilon_0\gamma 3\sqrt{3}\|S\| \text{ и } \frac{1}{\rho}\|S_1 - S_2\| \leq \varepsilon_1\sqrt{3}\|S\|. \tag{37}$$

Итоговую оценку ϵ_S абсолютной погрешности вычисления двусторонней последовательности Штурма (а тем самым и отношений компонент собственного вектора) получаем суммированием (36) и (37) :

$$\|S - \frac{1}{\rho}\widetilde{S}_2\| \leq \widetilde{\epsilon}_S\|S\| = \epsilon_S,$$

где $\widetilde{\epsilon}_S$ – относительная погрешность

$$\widetilde{\epsilon}_S = \varepsilon_0\gamma 3\sqrt{3} + \varepsilon_1\sqrt{3} + 48\varepsilon_1\gamma\sqrt{3} = \left(\varepsilon_0 + \varepsilon_1\left(\frac{1}{3\gamma} + 16\right)\right)3\sqrt{3}\gamma. \tag{38}$$

3.5 Арифметика вынесенных порядков

Вычисление собственных векторов для матриц достаточно больших размеров может приводить к ситуациям ПЕРЕПОЛНЕНИЯ и ПОТЕРИ ПОРЯДКА. Для того, чтобы их избежать, необходимо искусственно увеличить точность вычисления в тех процедурах, где наиболее вероятен неконтролируемый рост вычислительных погрешностей. Напомним, что произвольное вещественное число x можно представить в специальном виде

$$x = \pm y\gamma^k, \tag{39}$$

где γ – фиксированное целое число (основание арифметики), целое число k – γ-ичный порядок числа x, вещественное число y – γ-ичная мантисса

$$\frac{1}{\gamma} \leq y < 1 \tag{40}$$

Увеличение точности машинных операций означает увеличение массива машинной памяти, отводимой для размещения мантисс и порядков вещественных чисел. Это можно делать за счет возможностей современных языков программирования, позволяющих определять необходимые пользовательские типы данных (см. [9]), или при помощи арифметики вынесенных порядков (АВП). В ее основу положена простая идея раздельного хранения порядка числа и его мантиссы. Это означает, что вещественное число заменяется на каноническую пару чисел: целое k – порядок и вещественное y – мантисса. Для этого удобно ввести специальные операции определяющие мантису и порядок числа x:

$$y = \mathrm{Fr}(x), \qquad k = \mathrm{Ex}(x) \tag{41}$$

Им соответствуют встроенные функции ФОРТРАНА-90, оперирующие с машинными числами, $FRACTION(x)$ и $EXPONENT(x)$ такие, что если если x – машинное вещественное число, то

$$FRACTION(x) = \mathrm{Fr}(x), \qquad EXPONENT(x) = \mathrm{Ex}(x).$$

Таким образом, машинное выделение мантиссы и порядка производится точно, без погрешностей.

Стандартные арифметические операции также допускают работу с каноническими парами. Пусть, например, x_1 и x_2 – два вещественных числа. Чтобы не обращать внимания на знак, будем считать, что оба они положительные: $x_1 > 0$ и $x_2 > 0$. Тогда согласно (39)

$$x_1 = y_1\gamma^{k_1}, \qquad x_2 = y_2\gamma^{k_2}.$$

Приведем очевидные соотношения

$$x_1 \times x_2 = y_1 \times y_2 \times \gamma^{k_1+k_2}, \qquad x_1/x_2 = y_1/y_2 \times \gamma^{k_1-k_2}.$$

Выражения для сложения и вычитания чисел, представленных каноническими парами, не столь элементарны и однозначны. Поэтому арифметику вынесенных порядков удобно использовать в тех участках программ, где производится много умножений и делений, например при вычислении компонент собственного вектора по известной двусторонней последовательности Штурма (см. формулу (26)).

Оценим вычислительные погрешности, возникающие при умножении и делении машинных чисел, представленных каноническими парами. Для этого отметим, что вычисление суммы и разности порядков k_1+k_2 и k_1-k_2 производится точно в достаточно большом диапазоне. В то же время умножение и деление мантисс y_1 и y_2 производится с относительной точностью вследствии неравенства (40) и леммы 2 [3]:

$$(y_1 \times y_2)_{\text{маш}} = (y_1 \times y_2)(1+\alpha), \qquad (y_1/y_2)_{\text{маш}} = (y_1/y_2)(1+\beta),$$

где $|\alpha|, |\beta| \le \varepsilon_1$. Будем представлять результат операции также в виде канонической пары, иными словами будем считать, что умножение на степень γ производится точно. Тогда

$$\begin{aligned} (y_1 \times y_2)_{\text{маш}} \times \gamma^{k_1+k_2} &= (x_1 \times x_2)(1+\alpha) = Fr((y_1 \times y_2)(1+\alpha)) \times \gamma^{Ex((y_1\times y_2)(1+\alpha))+k_1+k_2}, \\ (y_1/y_2)_{\text{маш}} \times \gamma^{k_1-k_2} &= (x_1/x_2)(1+\beta) = Fr((y_1/y_2)(1+\beta)) \times \gamma^{Ex((y_1/y_2)(1+\alpha))+k_1-k_2}. \end{aligned} \tag{42}$$

Это означает, что данные вычислительные операции над каноническими парами производятся с относительной точностью, причем с максимальной, что является одним из основных преимуществ арифметики вынесенных порядков, так как упрощает анализ погрешностей.

Пусть теперь нам известны компоненты канонической пары: порядок k и мантисса y. Для того, чтобы восстановить по ним соответствующее вещественное число, необходимо произвести возведение в степень, умножение согласно формуле (39) и учесть знак. Если эти действия производятся при помощи компьютера, то погрешности результата могут быть проанализированы следующим образом: возведение основания арифметики γ в целую степень производится точно, а умножение мантиссы на степень γ – с абсолютной погрешностью

$$(y \times \gamma^k)_{\text{маш}} = y\,\gamma^k + \xi, \tag{43}$$

причем умножение на положительную степень γ производится точно: $\xi = 0$, а погрешности при умножении на отрицательную степень γ не превосходят ε_0: $|\xi| \le \varepsilon_0$.

3.6 Вычисление компонент собственного вектора трёхдиагональной симметричной матрицы с гарантированной оценкой точности результата.

Предварительные замечания

Пусть T – квадратная трехдиагональная матрица (не обязательно симметричная) размера M

$$T = \begin{pmatrix} \widetilde{d}_1 & \widetilde{b}_2 & & & & 0 \\ \widetilde{c}_2 & \widetilde{d}_2 & \widetilde{b}_3 & & & \\ & \widetilde{c}_3 & \widetilde{d}_3 & \widetilde{b}_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \widetilde{c}_{M-1} & \widetilde{d}_{M-1} & \widetilde{b}_M \\ 0 & & & & \widetilde{c}_M & \widetilde{d}_M \end{pmatrix}.$$

Потребуем, чтобы все элементы главной и побочных диагоналей были отличны от нуля. Обозначим λ_n – собственное значение матрицы T. Справедлива следующая лемма.

Лемма 2. *Последовательность $\mathcal{P}_j = \mathcal{P}_j(\lambda_n)$, вычисленная по правилу*

$$\mathcal{P}_0 = 0,$$

$$\mathcal{P}_j = \frac{|\widetilde{b}_{j+1}|}{\widetilde{d}_j - \lambda_n - |\widetilde{c}_j|\mathcal{P}_{j-1}} \quad (j = 1, 2, \ldots M-1),$$

является двусторонней, причем среди ее элементов с номерами $j = 1, 2, \ldots, M-1$ нет нулевых и бесконечных:

$$0 < |\mathcal{P}_j| < \infty, \qquad j = 1, 2, \ldots, M-1$$

Пусть

$$v(T) = (v_1, v_2, \ldots, v_M)^T$$

есть собственный вектор матрицы T, соответствующий собственному числу λ_n. По аналогии с формулой (26), можно установить, что

$$v_{j+1} = -\mathrm{sign}(\widetilde{b}_{j+1})\frac{v_j}{\mathcal{P}_j} \tag{44}$$

Как правило для удобства практического использования собственные векторы нормируют специальным образом. Мы будем искать собственный вектор с единичной нормой: $\|v(T)\| = 1$. Чтобы определить компоненты нормированного вектора, достаточно вычислить компоненты любого вектора, пропорцонального v: $u = Cv$, а затем нормировать его $v = u/\|u\|$ (таким образом коэффициент пропорциональности C равен $\|u\|$). Заметим, что компоненты произвольно нормированного вектора u также удовлетворяют соотношениям (44):

$$u_{j+1} = -\mathrm{sign}(\widetilde{b}_{j+1})\frac{u_j}{\mathcal{P}_j}.$$

дополним их равенством

$$u_1 = 1.$$

Это позволит нам вычислить все остальные компоненты u по индукции.

Эти формулы легко модифицировать так, что сначала определяются модули компонент

$$\begin{gathered} |u_1| = 1, \\ |u_{j+1}| = |u_j| \;/\; |\mathcal{P}_j| \quad (1 \le j \le M-1)\,, \end{gathered} \tag{45}$$

а затем их знаки:

$$\begin{gathered} \mathrm{sign}(u_1) = 1, \\ \mathrm{sign}(u_{j+1}) = -\mathrm{sign}(\widetilde{b}_{j+1})\mathrm{sign}(u_j)\mathrm{sign}(\mathcal{P}_j) \quad (1 \le j \le M-1)\,, \end{gathered} \tag{46}$$

Представим компоненты вектора u и элементы двусторонней последовательности Штурма в виде канонических пар (воспользовавшись для этого операциями (41)) и перепишем соотношения (45) в терминах соответствующих мантисс $Fr(u_j)$, $Fr(\mathcal{P}_j)$ и порядков $Ex(u_j)$, $Ex(\mathcal{P}_j)$:

$$Fr(u_1) = \frac{1}{\gamma}, \qquad Ex(u_1) = 1, \tag{47}$$

$$Fr(u_{j+1}) = Fr\left(\frac{Fr(u_j)}{Fr(\mathcal{P}_j)}\right) \quad (1 \le j \le M-1)\,, \tag{48}$$

$$Ex(u_{j+1}) = Ex\left(\frac{Fr(u_j)}{Fr(\mathcal{P}_j)}\right) + Ex(u_j) - Ex(\mathcal{P}_j) \quad (1 \le j \le M-1)\,. \tag{49}$$

На основе этих выкладок организуется вычисление мантисс и порядков модулей компонент вектора u. Для окончательного определения компонент необходимо учесть их знак согласно формуле (46).

Напомним, что собственный вектор $v(T)$ с единичной нормой может быть получен из вектора u при помощи нормировки. Мы будем нормировать вектор u в два шага. Сначала вычислим максимальный порядок компонент u_j

$$K = \max_j\{Ex(u_j)\}.$$

После этого максимальный порядок вектора $u\gamma^{-K}$ не превосходит нуля. Вынесение максимального порядка и замена вектора u на коллинеарный ему вектор $u^{(K)} = u\gamma^{-K}$ делается для того, чтобы при вычислении компонент и нормы вектора не возникало ситуаций ПЕРЕПОЛНЕНИЯ. Затем вычисляется евклидова норма получившегося вектора

$$U = \sqrt{\sum_{j=1}^{M}\left(u_j^{(K)}\right)^2}.$$

Так как по построению максимальный порядок компонент вектора $u^{(K)}$ равен нулю, то верна следующая оценка:

$$U \geq \max_j\{|u_j^{(K)}|\} \geq \frac{1}{2}. \tag{50}$$

Компонентам искомого вектора v с учетом знака присваиваются следующие значения

$$v_1 = \frac{u_1^{(K)}}{U}, \tag{51}$$

$$v_{j+1} = \frac{\mathrm{sign}(u_{j+1})u_{j+1}^{(K)}}{U} \qquad j = 1, \ldots, M-1. \tag{52}$$

Евклидова норма, определенного таким образом собственного вектора v равна 1.

Алгоритм вычисления нормированного собственного вектора трехдиагональной симметричной матрицы с оценкой точности результата.

Дано: трехдиагональная симметричная матрица S размера M,
n – номер собственного значения матрицы $1 \leq n \leq M$ (собственные значения матрицы считаются занумерованными по возрастанию).

Шаг 1. Вычисление двусторонней последовательности Штурма.
Вычислить приближенную двустороннюю последовательность Штурма $\mathcal{P}_{j\text{маш}}$ ($j = 1, \ldots, M-1$) для n – го собственного значения трехдиагональной симметричной матрицы S, лежащего в гарантированнх границах $\lambda_n^{(+)} \leq \lambda_n \leq \lambda_n^{(-)}$, по алгоритму описанному в разделе 3.4:

$$\mathcal{P}_{j\text{маш}} = \mathcal{P}_{j\text{маш}}^{(+)}(\lambda_n^{(+)}) \text{ при } 1 \leq j \leq J-1, \qquad \mathcal{P}_{j\text{маш}} = \mathcal{P}_{j\text{маш}}^{(-)}(\lambda_n^{(-)}) \text{ при } J \leq j \leq M-1$$

Замечание. Согласно лемме 1 и оценке (38) последовательность $\mathcal{P}_{j\text{маш}}$ является точной двусторонней последовательностью для некоторой трехдиагональной, вообще говоря, несиметричной матрицы $T = \frac{1}{\rho}\widetilde{S}_2$, которая по норме мало отличается от исходной матрицы S (см. далее оценки погрешностей).

Разделить элементы последовательности Штурма на мантиссу и порядок:

$$\mathcal{Q}_j = Fr(\mathcal{P}_{j\text{маш}}), \qquad s_j = Ex(\mathcal{P}_{j\text{маш}}) \qquad j = 1, \ldots, M-1$$

Шаг 2. Определение мантисс компонент приближенного собственного вектора.
Присвоить

$$\mu_1 = \frac{1}{\gamma}.$$

Вычислить последовательно остальные мантиссы используя рекуррентные соотношения

$$\mu_{j+1} = Fr(\mu_j \oslash \mathcal{Q}_j) \qquad j = 1, \ldots, M-1.$$

Шаг 3. Определение порядков компонент приближенного собственного вектора.

Определить порядки компонент приближенного собственного вектора из следующих рекуррентных соотношений

$$t_1 = 1,$$

$$t_{j+1} = t_j + Ex(\mu_j \oslash \mathcal{Q}_j) - s_j, \qquad j = 1, \ldots, M-1,$$

где $\mathcal{Q}_j$ и s_j – мантиссы и порядки элементов последовательности Штурма.

Шаг 4. Нормировка вектора с вынесением максимального порядка.

Найти максимальный порядок компонент вектора:

$$\widetilde{K} = \max_j \{t_j\}$$

Вычислить модули компонент и евклидову норму вектора $\widetilde{u}^{(K)}$

$$|\widetilde{u}_j^{(K)}| = \mu_j \times \gamma^{t_j - \widetilde{K}}, \qquad \widetilde{U} = \sqrt{\sum_{j=1}^{M} \left(\widetilde{u}_j^{(K)}\right)^2}$$

Найти модули приближенных значений нормированных компонент $\widetilde{v}_j$ искомого нормированного собственного вектора согласно следующим формулам:

$$|\widetilde{v}_j| = |\widetilde{u}_j| / \widetilde{U}.$$

Определить их знак

$$\mathrm{sign}(\widetilde{v}_1) = 1,$$

$$\mathrm{sign}(\widetilde{v}_{j+1}) = -\mathrm{sign}(b_{j+1})\mathrm{sign}(\widetilde{v}_j)\mathrm{sign}(\mathcal{P}_{j\text{маш}}) \qquad j = 1, \ldots, M-1.$$

Замечание. В этой формуле можно использовать $\mathrm{sign}(b_{j+1})$ знак элемента исходной матрицы, так как все преобразования, переводящие матрицу S в T, сохраняют знаки элементов побочной диагонали (см. **Шаг 2** алгоритма вычисления двусторонней последовательности Штурма и равенства (33)).

Шаг 5. Оценка погрешности результата.

Определить абсолютную погрешность ϵ_{SV} по формуле (64) или относительную $\widetilde{\epsilon}_{SV}$ по формуле (65).

Результатом выполнения алгоритма является вектор $\widetilde{v}(S) = (\widetilde{v}_1, \widetilde{v}_2, \ldots, \widetilde{v}_M)^T$ – приближенное значение точного нормированного собственного вектора $v(S)$ трехдиагональной симметричной матрицы S ($\|v(S)\| = 1$) в смысле следующего равенства:

$$\widetilde{v}(S) = v(S + \triangle S) + \triangle v = v(T) + \eta, \tag{53}$$

где для близкой к S матрицы T и вектора погрешности η верна следующая оценка

$$\|S - T\| + \|\eta\| \le \widetilde{\epsilon}_{SV} \|S\| = \epsilon_{SV}.$$

Оценка погрешностей

Вычислительные погрешности, накапливаемые на первых двух шагах алгоритма при вычислении двусторонней последовательности Штурма уже оценены ранее и трактуются как оценка нормы разности между исходной матрицей S и некоторой матрицей T, для которой вычисленная приближенная двусторонняя последовательность Штурма $\mathcal{P}_{j\text{маш}}$ является точной.

Шаги 2, 3 и **4** алгоритма посвящены вычислению компонент нормированного собственного вектора матрицы T, которая мало отличается от S. Оценим возникающие при этом погрешности. Для этого мы сравним результаты **Шагов 2, 3, 4** с формулами (47)-(52)

Заметим, что согласно (47)

$$|u_1| = \mu_1 \gamma^{t_1},$$

таким образом вычисление первой компоненты ненормированного собственного вектора u как канонической пары (т.е. без рассмотрения точности умножения на степень γ) производится точно.

Рассмотрим подробнее процесс вычисления второй компоненты. Так как машинное деление мантисс производится с относительной точностью, а все остальные операции – точно, то, используя формулы (42), можно убедится, что

$$\mu_2 \gamma^{t_2} = (1+\xi_2) Fr(u_2) \gamma^{Ex(u_2)} = (1+\xi_2)|u_2|,$$

где $|\xi_2| \le \varepsilon_1$.

Для того, чтобы оценить погрешность вычисления мантиссы и порядка компоненты u_3, нужно сравнить $\mu_3 \gamma^{t_3}$ и $(1+\xi_2)|u_3|$. При этом необходимо рассмотреть оба числа как канонические пары и воспользоваться равенствами, аналогичными (48) и (49):

$$Fr((1+\xi_2)u_3) = Fr\left(\frac{Fr((1+\xi_2)u_2)}{\mathcal{Q}_2}\right),$$

$$Ex((1+\xi_2)u_3) = Ex\left(\frac{Fr((1+\xi_2)u_2)}{\mathcal{Q}_2}\right) + Ex((1+\xi_2)u_2) - s_2.$$

В результате такого сравнения получим

$$\mu_3 \gamma^{t_3} = (1+\xi_3)(1+\xi_2)|u_3|, \qquad |\xi_3| \le \varepsilon_1$$

Очевидно, что накопление погрешностей при вычислении последующих компонент происходит таким же образом и, продолжая их анализ по индукции, приходим к следующему выражению

$$\mu_j \gamma^{t_j} = \prod_{k=2}^{j} (1+\xi_k)|u_j|, \qquad |\xi_k| \le \varepsilon_1.$$

Выведем очевидную оценку, которая понадобится впоследствии:

$$\begin{gathered}\textstyle\prod_{k=2}^{j}(1+\xi_k) - 1 \le (1+\varepsilon_1)^{M-1} - 1 = \exp((M-1)\ln(1+\varepsilon_1)) - 1 \le \exp((M-1)\varepsilon_1) - 1 \le \\ 1 + \varepsilon_1 (M-1)\exp(\varepsilon_1(M-1)) - 1 \le 2\varepsilon_1(M-1)\end{gathered} \tag{54}$$

Последнее неравенство в цепочке выполнено при условии, если выполнено неравенство $M \le \frac{\ln 2}{\varepsilon_1} + 1$, что, как правило, имеет место на практике.

Следующий ряд простых неравенств также понадобится для вывода окончательной оценки. Пусть верно неравенство

$$|a-b| \le \delta |a|.$$

Тогда очевидно

$$|b| \le (1+\delta)|a|, \qquad \left|\frac{|a|}{|b|} - 1\right| \le \delta, \qquad \frac{1-\delta}{|a|} \le \frac{1}{|b|} \le \frac{1+\delta}{|a|}. \tag{55}$$

Далее легко установить, что

$$\left|\frac{1}{|a|} - \frac{1}{|b|}\right| = \frac{|a-b|}{|a|\,|b|} \le \frac{\delta|a|}{|a|\,|b|} \le \frac{\delta}{|b|} \le \frac{\delta(1+\delta)}{|a|}. \tag{56}$$

Так как по определению

$$\widetilde{u}_j^{(K)} = (\mu_j \times \gamma^{t_j - \widetilde{K}})_{\text{маш}},$$

то согласно (43) и (54) имеем оценку

$$|u_j^{(K)} - \widetilde{u}_j^{(K)}| \le 2\varepsilon_1 (M-1)|u_j^{(K)}| + \varepsilon_0.$$

Переходя от компонент к векторам, находим

$$\|u^{(K)} - \widetilde{u}^{(K)}\| \le 2\varepsilon_1(M-1)\|u^{(K)}\| + \varepsilon_0\sqrt{M}. \tag{57}$$

Для простоты записи обозначим $\delta_1 = 2(\varepsilon_1(M-1)+\varepsilon_0\sqrt{M})$. Воспользовавшись тем, что $\|u^{(K)}\| = U$, а также неравенствами (56) и (50), получим

$$\left|\frac{1}{U} - \frac{1}{\|\widetilde{u}^{(K)}\|}\right| \le \delta_1(1+\delta_1)\frac{1}{U}. \tag{58}$$

Из неравенств (55) и (57) следует

$$\frac{1}{\|\widetilde{u}^{(K)}\|} \le (1+\delta_1)\frac{1}{U}. \tag{59}$$

Далее вычисляется норма $\widetilde{U} = \|\widetilde{u}^{(K)}\|$. Обозначим $\widetilde{U}_{\text{маш}}$ результат выполнения этой операции на ЭВМ. Воспользуемся оценкой (20) [3], из которой следует, что

$$|\widetilde{U}_{\text{маш}} - \widetilde{U}| \le \varepsilon_1\frac{M+4}{2}\widetilde{U} = \delta_2\widetilde{U}, \qquad \delta_2 = \varepsilon_1\frac{M+4}{2}.$$

Неравенства (55), (56), (58) приводят к следующему результату

$$\left|\frac{1}{\widetilde{U}_{\text{маш}}} - \frac{1}{\widetilde{U}}\right| \le \delta_2(1+\delta_2)\frac{1}{\widetilde{U}} \le \delta_2(1+\delta_1)(1+\delta_2)\frac{1}{U}. \tag{60}$$

и

$$\frac{1}{\widetilde{U}_{\text{маш}}} \le (1+\delta_2)\frac{1}{\widetilde{U}} \le (1+\delta_1)(1+\delta_2)\frac{1}{U}. \tag{61}$$

Последний шаг алгоритма – нормировка вектора, которая производится покомпонентно, при этом

$$\widetilde{v}_j = \left(\widetilde{u}_j^{(K)}/\widetilde{U}\right)_{\text{маш}} \text{ и } \left|\widetilde{v}_j - \frac{\widetilde{u}_j^{(K)}}{\widetilde{U}}\right| \le \varepsilon_1\left|\frac{\widetilde{u}_j^{(K)}}{\widetilde{U}}\right| + \varepsilon_0$$

или

$$\left\|\widetilde{v} - \frac{1}{\widetilde{U}}\widetilde{u}^{(K)}\right\| \le \varepsilon_1\frac{\|\widetilde{u}^{(K)}\|}{\widetilde{U}} + \varepsilon_0\sqrt{M} \tag{62}$$

Следующее неравенство является следствием неравенства треугольника

$$\|\widetilde{v} - v\| \le \left\|\widetilde{v} - \frac{\widetilde{u}^{(K)}}{\widetilde{U}_{\text{маш}}}\right\| + \left\|\frac{\widetilde{u}^{(K)}}{\widetilde{U}_{\text{маш}}} - \frac{\widetilde{u}^{(K)}}{\widetilde{U}}\right\| + \left\|\frac{\widetilde{u}^{(K)}}{\widetilde{U}} - \frac{\widetilde{u}^{(K)}}{U}\right\| + \left\|\frac{\widetilde{u}^{(K)}}{U} - v\right\|$$

Простые преобразования с учетом (62) приводят к оценке

$$\|\widetilde{v} - v\| \le \varepsilon_1\frac{\|\widetilde{u}^{(K)}\|}{\widetilde{U}_{\text{маш}}} + \varepsilon_0\sqrt{M} + \|\widetilde{u}^{(K)}\|\left|\frac{1}{\widetilde{U}_{\text{маш}}} - \frac{1}{\widetilde{U}}\right| + \|\widetilde{u}^{(K)}\|\left\|\frac{1}{\widetilde{U}} - \frac{1}{U}\right\| + \frac{1}{U}\left\|\widetilde{u}^{(K)} - v\right\|.$$

Применим последовательно к каждому слагаемому неравенства (55) – (61). В результате с учетом того, что $\|u^{(K)}\| = U$, мы получаем итоговую оценку.

$$\|\widetilde{v} - v\| \le \varepsilon_1(1+\delta_1)^2(1+\delta_2) + \varepsilon_0\sqrt{M} + \delta_2(1+\delta_1)^2(1+\delta_2) + \delta_1(1+\delta_1)^2 + \delta_1 \le$$

$$4\max\{\delta_1,\delta_2\}(1+\delta_1)^2(1+\delta_2) + \varepsilon_0\sqrt{M}.$$

Неравенство (63) доказано.

Таким образом мы показали, что вычисленный вектор $\widetilde{v}$ есть возмущение точного собственного вектора некоторой матрицы T, близкой к исходной матрице S (53). Норма возмущения матрицы подчиняется оценке

$$\|S - T\| \le \widetilde{\epsilon}_S\|S\| = \epsilon_S,$$

причем относительная $\widetilde{\epsilon}_S$ и абсолютная ϵ_S погрешности определяются по формуле (38). Вектор погрешности η оценивается абсолютной величиной

$$\|\eta\| \le \epsilon_V(S),$$

где

$$\epsilon_V(S) = 4\max\{\delta_1, \delta_2\}(1+\delta_1)^2(1+\delta_2) + \varepsilon_0\sqrt{M} \tag{63}$$

при том, что $\delta_1 = 2(\varepsilon_1(M-1) + \varepsilon_0\sqrt{M})$, $\delta_2 = \varepsilon_1\frac{M+4}{2}$.

В качестве выходного параметра для алгоритма вычисления собственного вектора удобно использовать одну величину, которая оценивала бы и погрешность, внесенную в исходную матрицу, и вектор возмущения η. Такой величиной может быть, например, сумма этих абсолютных погрешностей:

$$\epsilon_{SV}(S) = \epsilon_S + \epsilon_V(S), \tag{64}$$

или относительная погрешность

$$\widetilde{\epsilon}_{SV}(S) = \widetilde{\epsilon}_S + \epsilon_V(S)\frac{\sqrt{3}}{\mathcal{M}(S)}. \tag{65}$$

3.7 Вычисление собственных значений и собственных векторов произвольной симметричной матрицы с гарантированной оценкой точности результата

Вычисление собственных значений и векторов произвольных симметричных матриц сводится к обработке симметричных матриц при помощи процедуры трехдиагонализации. Действительно, любая симметричная матрица A размера M ортогональными преобразованиями приводится к трехдиагональной форме $S = PAP^*$ (см. раздел 3.1). Из цепочки равенств (2) следует, что n – ое собственное значение $\lambda_n(A)$ матрицы A совпадает с n – м собственным значением $\lambda_n(S)$ трехдиагональной симметричной матрицы S: $\lambda_n(A) = \lambda_n(S)$, а соответствующий собственный вектор $v(A)$ равен произведению транспонированной матрицы преобразования P^T на собственный вектор $v(S)$ матрицы S: $v(A) = P^T v(S)$.

Алгоритм вычисления собственного значения произвольной симметричной матрицы с оценкой точности результата.

Дано: произвольная симметричная матрица A размера M,
n – номер собственного значения матрицы $1 \le n \le M$.

Шаг 1. Трехдиагонализация матрицы A.
Вычислить трехдиагональную матрицу S такую, что $S = PAP^*$ (алгоритм раздела 3.1) и абсолютную погрешность трехдиагонализации ϵ_T по формуле (13).

Шаг 2. Вычисление собственного значения.
Вычислить приближенную величину $\widetilde{\lambda}_n(S)$ n – го собственного значения $\lambda_n(S)$ трехдиагональной симметричной матрицы S (раздел 3.2) и абсолютную погрешностью $\epsilon_\lambda(S)$ по формуле (18). Присвоить $\widetilde{\lambda}_n(A) = \widetilde{\lambda}_n(S)$.

Шаг 3. Оценка погрешности результата.
Абсолютная погрешность вычисления n – го собственного значения матрицы A оценивается суммой погрешности трехдиагонализации ϵ_T (13) и погрешности вычисления собственного значения $\epsilon_\lambda(S)$ (18) матрицы S:

$$|\widetilde{\lambda}_n(A) - \lambda_n(A)| \le \epsilon_\lambda(A) = \epsilon_T + \epsilon_\lambda(S)$$

Относительная погрешность оценивается отношением $(\epsilon_T + \epsilon_\lambda)/\|S\|$.

Результатом выполнения алгоритма является приближенная величина $\widetilde{\lambda}_n(A)$ собственного значения $\lambda_n(A)$ и гарантированные границы $\lambda_n^{(-)}(A)$ и $\lambda_n^{(+)}(A)$ точного собственного значения $\lambda_n(A)$:

$$\lambda_n^{(-)}(A) = \widetilde{\lambda}_n(A) - \epsilon_\lambda(A) \le \lambda_n(A) \le \widetilde{\lambda}_n(A) + \epsilon_\lambda(A) = \lambda_n^{(+)}(A)$$

Алгоритм вычисления нормированного собственного вектора произвольной симметричной матрицы с оценкой точности результата.

Обязательные данные: произвольная симметричная матрица A размера M,
n – номер собственного значения матрицы $1 \le n \le M$.

Необязательные данные: гарантированные нижняя $\lambda_n^{(-)}$ и верхняя $\lambda_n^{(+)}$ ($\lambda_n^{(-)} \le \lambda_n^{(+)}$)границы n-го собственного значения λ_n матрицы A

Шаг 1. Трехдиагонализация матрицы A.
Вычислить трехдиагональную матрицу S такую, что $S = PAP^*$, матрицу ортогонального преобразования P (алгоритм раздела 3.1) и абсолютную погрешность трехдиагонализации ϵ_T по формуле (13).

Шаг 2. Дополнительные вычисления.
Если границы $\lambda_n^{(-)}$, $\lambda_n^{(+)}$ заданы и $\lambda_n^{(-)} < \lambda_n^{(+)}$, то перейти к **Шагу 3.**

Если не известны границы $\lambda_n^{(-)}$, $\lambda_n^{(+)}$, вычислить их, исходя из полученной на **Шаге 1.** трехдиагональной симметричной матрицы S, по алгоритму раздела 3.2.

Если границы заданы, но совпадают $\lambda_n^{(-)} = \lambda_n^{(+)} = \lambda$, то переопределить их следующим образом $\lambda_n^{(-)} := \lambda - \epsilon_\lambda(S)$, $\lambda_n^{(+)} := \lambda + \epsilon_\lambda(S)$, где $\epsilon_\lambda(S)$ определяется формулой (18).

Шаг 3. Вычисление собственного вектора матрицы S.
Вычислить вектор $v(S)$, соответствующий $\lambda_n(S) = \lambda_n(A)$, применяя алгоритм раздела 3.6. В результате будет получен вектор $\widetilde{v}(S)$, близкий к $v(S)$ в смысле соотношения (53)

Шаг 4. Вычисление собственного вектора матрицы A.
Вычислить собственный вектор матрицы A по формуле $\widetilde{v}(A) = (P^T \times \widetilde{v}(S))_{\text{маш}}$.

Шаг 5. Оценка погрешности результата.
Вычислить абсолютную ϵ_{TSV} или относительную $\widetilde{\epsilon}_{TSV}$ погрешность по формулам

$$\epsilon_{TSV}(A) = \epsilon_T + \epsilon_S + \epsilon_V(S) + \varepsilon_1(M+1)\sqrt{M}, \qquad \widetilde{\epsilon}_{TSV} = \epsilon_{TSV}(A)\frac{\sqrt{3}}{\mathcal{M}(S)},$$

используя (13), (38), (63).

Замечание. Последнее слагаемое возникает при умножении матрицы P^T на вектор $\widetilde{v}(S)$ (см. [3] стр. 28).

Результатом выполнения алгоритма является вектор $\widetilde{v}(A) = (v_1, v_2, \ldots, v_M)^T$ –приближенное значение точного нормированного собственного вектора $v(A)$ матрицы A в смысле следующего равенства:

$$\widetilde{v}(A) = v(A + \triangle A) + \triangle v = v(B) + \zeta,$$

где для матрицы B близкой к матрице A и вектора погрешности ζ, верна оценка

$$\|A - B\| + \|\zeta\| \le \epsilon_{TSV}(A).$$

4 Обзор возможностей пакета GALA

Рассмотренные выше алгоритмы были использованы при разработке программ для решения симметричных спектральных задач линейной алгебры. Эти программы объединены в модуль Symmod, который дополнил пакет программ GALA (Guaranted Accuracy in Linear Algebra), описанный ранее в [3]. В ходе работы были внесены изменения и в другие ранее написанные модули, поэтому в Приложение мы поместили полное подробное описание всех модулей. Программы написаны на языке FORTRAN-90 (с использованем формата двойной точности) и работают только с вещественными матрицами.

Основная отличительная черта пакета GALA наличие гарантированной оценки точности результата, которая присутствует во всех процедурах в качестве необязательного параметра.

Коротко остановимся на некоторых подпрограммах этого пакета. Для работы с произвольными прямоугольными матрицами основными являются две процедуры. Первая – *LinSystemSolution* – решает систему $Ax + r = f$ с произвольной прямоугольной $N \times M$-матрицей A. Ее параметры: матрица A, правая часть f, решение x, невязка r и оценка точности решения.

Вторая – *InverseMatrix* – производит обращение квадратной матрицы. В параметры этой процедуры, кроме исходной и обратной матриц, входит также оценка ошибки обращения. Элементы обратной матрицы могут быть уточнены методом итераций с помощью процедуры *RefineInverseMatrix*

Результатами процедур *SingValue*, *TwoDiagCond* являются соответственно сингулярное число и обусловленность квадратной двухдиагональной матрицы, а *MatCond* – число обусловленности произвольной матрицы.

Кроме этого, пакет содержит ряд вспомогательных служебных процедур, некоторые из которых могут заинтересовать подготовленного пользователя. Например, подпрограмма *TwoDiagonalisation* с помощью ортогональных преобразований приводит произвольную $N \times M$-матрицу A $(N \geq M)$ к двухдиагональному виду.

Основными процедурами нового модуля Symmod, работающего с симметричными матрицами, можно считать *EigenValueSymMatrix* (вычисляет j – е собственное значение) и *EigenVectorSymMatrix* (вычисляет компоненты соответствующего нормированного собственного вектора).

При работе с симметричными матрицами могут оказаться полезными также следующие вспомогательные процедуры. Функция *SymMatCond* вычисляет число обусловленности симметричной матрицы. Подпрограмма *ThreeDiagonalization* с помощью ортогональных преобразований приводит произвольную симметричную матрицу к трехдиагональному виду. Для трехдиагональной матрицы процедура *EigenValueThreeDSMatrix* вычисляет j – е собственное значение, а *EigenVectorThreeDSMatrix* – компоненты соответствующего нормированного собственного вектора. Функция *ThreeDiagCond* выдает число обусловленности трехдиагональной симметричной матрицы, а *NormThreeDSMatrix* вычисляет оценку сверху ее евклидовой нормы.

4.1 Пример

Обсудим на примере упомянутые во введении этапы численного исследования физических процессов и заодно продемонстрируем возможности пакета GALA.

Одной из самых распространенных спектральных физических задач является нахождение общей энергии частицы E и ее волновой функции ψ из уравнения Шредингера

$$\Delta\psi + \frac{2\mu}{\hbar^2}(E - U(x, y, z))\psi = 0,$$

где $\hbar$ – постоянная Планка, μ – масса частицы, а $U = U(x, y, z)$ – ее потенциальная энергия.

На практике функция U либо задана аналитически, либо ее значения известны только в некоторых точках и получены в ходе физического или численного эксперимента (и тогда величина U в произвольной точке определяется интерполяцией).

Рассмотрим частный вид уравнения Шредингера для гармонического осциллятора:

$$\frac{\hbar^2}{2\mu}\frac{d^2\psi}{dx^2} + (E - \frac{\mu\omega_0^2}{2}x^2)\psi = 0\,, \tag{66}$$

которое мы дополним естественным условием нормировки

$$\int_{-\infty}^{+\infty} |\psi(x)|^2 \, dx = 1, \tag{67}$$

Здесь ω_0^2 – собственная частота осциллятора.

В этом случае нет необходимости в определении исходных данных из эксперимента (этап 1, см. Введение) и их интерполяции (этап 2), так как масса частицы считается известной, а потенциальная энергия U задана явным аналитическим образом. Для численного анализа необходимо лишь построить дискретную модель и решить спектральную задачу (1). После обозначений

$$\lambda = \frac{2E}{\hbar\omega_0}, \qquad x_0 = \sqrt{\frac{\hbar}{\mu\omega_0}}, \qquad \xi = \frac{x}{x_0}, \qquad \varphi(\xi) = \psi(x_0\xi),$$

уравнение (66) превращается в обобщенное уравнение гипергеометрического типа с соответствующей нормировкой

$$\frac{d^2\varphi}{d\xi^2} + (\lambda - \xi^2)\varphi = 0, \qquad \int_{-\infty}^{+\infty} |\varphi(\xi)|^2 \, d\xi = \frac{1}{x_0}. \tag{68}$$

Отметим, что в данном простом случае собственные функции задачи (68) имеют аналитическое представление через полиномы Эрмита (см. [10]):

$$\varphi_n(\xi) = \frac{1}{\sqrt{x_0}} \frac{e^{-\frac{1}{2}\xi^2} H_n(\xi)}{\sqrt{2^n n! \sqrt{\pi}}}. \tag{69}$$

Им соответствуют собственные значения

$$\lambda_n = 2n + 1 \text{ при } n = 0, 1, 2 \ldots \tag{70}$$

Возвращаясь к исходным переменным, получаем

$$\psi_n(x) = \frac{1}{\sqrt{x_0}} \frac{e^{-\frac{1}{2}\left(\frac{x}{x_0}\right)^2} H_n\left(\frac{x}{x_0}\right)}{\sqrt{2^n n! \sqrt{\pi}}}, \qquad E_n = \hbar\omega_0 \left(n + \frac{1}{2}\right) \text{ при } n = 0, 1, 2 \ldots$$

Применим теперь к задаче (68) численные методы и сравним вклады в общую погрешность решения ошибки вычисления собственных значений и векторов матрицы дискретной модели и ошибки дискретизации. Существенно, что наличие точного аналитического решения (69), (70) позволит нам провести необходимое сравнение, не углубляясь в теорию разностных схем.

Дискретная модель. Для того, чтобы построить дискретную математическую модель задачи (68), зафиксируем длину шага Δ, введем функцию дискретного параметра $u^{(k)}$ и приблизим производную $\frac{d^2\varphi}{d\xi^2}$ конечной разностью

$$u^{(k)} = \varphi(k\,\Delta), \qquad \frac{d^2\varphi}{d\xi^2} \approx \frac{u^{(k-1)} - 2u^{(k)} + u^{(k+1)}}{\Delta^2}.$$

В результате вместо дифференциального уравнения (68) будет получена бесконечная система разностных уравнений

$$\frac{u^{(k-1)} - 2u^{(k)} + u^{(k+1)}}{\Delta^2} + (\lambda - (k\Delta)^2)u^{(k)} = 0. \tag{71}$$

Для того, чтобы получить конечную систему разностных уравнений, аппроксимирующую задачу (68), необходимо воспользоваться входящим в нее условием нормировки. Оно, в частности, означает, что функция $\varphi(\xi)$ на бесконечности близка к нулю. Следовательно, для некоторого большого N функции дискретного переменного $u^{(k)}$ и

$$v^{(k)} = \begin{cases} u^{(k)}, & |k| \le N, \\ 0, & |k| > N \end{cases}$$

близки:

$$u^{(k)} \approx v^{(k)} \tag{72}$$

и функция v также является дискретной аппроксимацией функции $\varphi(\xi)$. Значения функции v удовлетворяют следующим уравнениям:

$$\begin{array}{l} \frac{-2v^{(-N)}+v^{(-N+1)}}{\Delta^2} + (\lambda - (N\Delta)^2)v^{(N)} = 0 \\ \frac{v^{(k-1)}-2v^{(k)}+v^{(k+1)}}{\Delta^2} + (\lambda - (k\Delta)^2)v^{(k)} = 0, \qquad |k| < N-1 \\ \frac{v^{(N-1)}-2v^{(N)}}{\Delta^2} + (\lambda - (N\Delta)^2)v^{(N)} = 0. \end{array} \tag{73}$$

Ошибки дискретизации. Укажем на очевидные закономерности. Точность дискретной модели складывается из точности аппроксимации дифференциального уравнения и условия на бесконечности и, следовательно, зависит от двух параметров: шага Δ и числа N. Таким образом для уменьшения ошибки дискретизации необходимо одновременное уменьшение Δ и увеличение N. Отметим, что при достаточно большом N основную роль играет погрешность приближения дифференциального уравнения и ошибка дискретизации ведет себя как $O(\Delta^2)$.

Собственные значения и векторы. Матричный вид системы (73)

$$Sv = \lambda v. \tag{74}$$

Здесь вектор

$$v = (v^{(-N)}, v^{(-N+1)}, \ldots, v^{(k)}, \ldots, v^{(N-1)}, v^{(N)})^T,$$

а элементы матирицы S размера $(2N+1) \times (2N+1)$ определяются следующей формулой:

$$s_{ij} = s_{ji} = \begin{cases} 0, & j > i+1, \\ -1/\Delta^2, & j = i+1, \\ 2/\Delta^2 + (j-N+1)^2\Delta^2, & j = i, \end{cases} \qquad 1 \le j \le 2N+1.$$

Таким образом решение уравнения Шредингера свелось к нахождению собственных значений и векторов симметричной трехдиагональной матрицы S. Именно на этой стадии необходимо использование современных алгоритмов линейной алгебры с гарантированной оценкой точности результата. Напомним, что только в этом случае, как было указано во Введении, можно получить полное обоснование соответствия численного и аналитического решений. Используя процедуры *EigenValueThreeDSMatrix* и *EigenVectorThreeDSMatrix* пакета GALA для решения задачи (74), получаем приближенные величины $\widetilde{\lambda}_n$ и $\widetilde{v}_n^{(j)}$ первых собственных значений λ_n (70) и компонент собственных векторов v_n, которые разыскивались нами как аппроксимация собственных функций φ_n (69).

Анализ погрешностей. Выходные параметры *error* процедур *EigenValueThreeDSMatrix* и *EigenVectorThreeDSMatrix* пакета GALA представляют собой оценки относительных погрешностей $\widetilde{\epsilon}_\lambda(S)$ (19), $\widetilde{\epsilon}_{SV}(S)$ (65). Заметим, что согласно формулам (19), (65) относительные погрешности пропорциональны машинному числу ε_1 и практически не зависят от самой матрицы. Только относительная погрешность собственного вектора включает в себя зависимость от размера матрицы и фактически пропорциональна параметру дискретизации N.

Напомним, что абсолютные погрешности вычисления собственного значения и собственного вектора удовлетворяют неравенствам $\epsilon_\lambda(S) \le \widetilde{\epsilon}_\lambda(S)\mathcal{M}(S)$, см. формулу (18), и $\epsilon_{SV} \le \widetilde{\epsilon}_{SV}\mathcal{M}(S)$, формула (64). В нашем примере элементы матрицы S имеют простой вид и величина $\mathcal{M}(S)$ легко оценивается через параметры N и Δ:

$$\mathcal{M}(S) \le \frac{3}{\Delta^2} + (N+1)^2\Delta^2.$$

Таким образом, имеют место приближения

$$\epsilon_\lambda(S) \approx \varepsilon_1\mathcal{M}(S) \approx \varepsilon_1/\Delta^2, \qquad \epsilon_{SV}(S) \approx \varepsilon_1\mathcal{M}(S) \approx \varepsilon_1/\Delta^2.$$

Отсюда следует вывод, что погрешности вычисления собственных чисел и векторов растут с уменьшением Δ, в то время как погрешности дискретизации убывают. Существует такое значение

Δ, при котором они сравниваются. Например, если $\Delta = 10^{-4}$, а $\varepsilon_1 \approx 10^{-16}$, то и та и другая погрешности по порядку величины равны 10^{-8}. Это еще раз свидетельствует о том, что погрешностями вычислений пренебрегать нельзя.

Результаты вычислений. Введем величины $\delta_\lambda^{(n)} = |\widetilde{\lambda}_n - \lambda_n|$ и $\delta_v^{(n)} = \max_j |\widetilde{v}_n^{(j)} - \varphi_n(j\Delta)|$. Первая из них является общей погрешностью в определении собственных значений, а вторая – собственных функций. Они включают в себя как погрешность аппроксимации, так и погрешность вычислений.

Следующие графики иллюстрируют зависимость погрешностей $\delta_\lambda^{(n)}$, $\delta_v^{(n)}$, $\epsilon_\lambda(S)$ и ϵ_{SV} от значений Δ и n при постоянном $N = 3000$. Все вычисления производились на суперкомпьютере ALPHA с двойной точностью, то есть машинные константы принимали следующие значения:

$$\gamma = 2, \;\; \varepsilon_0 = 2.225E-308, \;\; \varepsilon_1 = 2.220E-16, \;\; \varepsilon_\infty = 1.798E+308$$

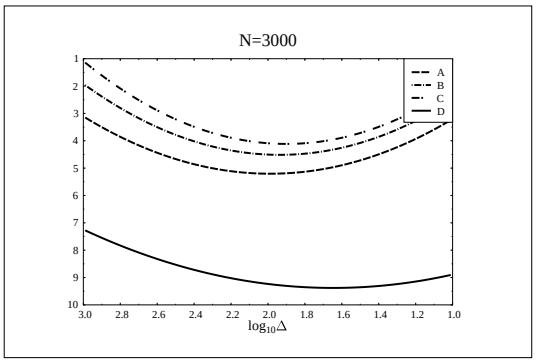

Рис.1

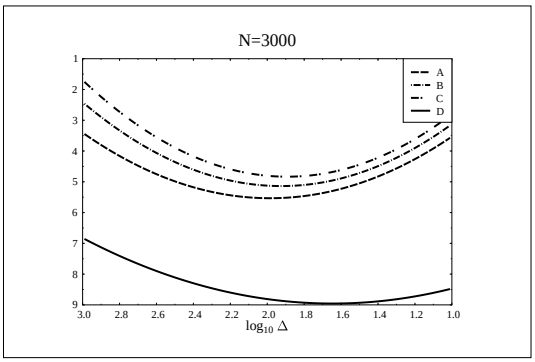

Рис.2

На рисунке 1 изображены следующие зависимости от десятичного порядка величины Δ: A – $\log_{10}\delta_\lambda^{(0)}$, B – $\log_{10}\delta_\lambda^{(1)}$, C – $\log_{10}\delta_\lambda^{(2)}$, D – $\log_{10}\epsilon_\lambda(S)$ при постоянном $N = 3000$.

На рисунке 2 приведены графики: A – $\log_{10}\delta_v^{(0)}$, B – $\log_{10}\delta_v^{(1)}$, C – $\log_{10}\delta_v^{(2)}$, D – $\log_{10}\epsilon_{SV}(S)$ также при $N = 3000$.

Комментарии. В первую очередь отметим, что график кривой D лежит существенно ниже других. Это значит, что в данном случае вклад арифметических погрешностей в общую погрешность вычисления собственных чисел незначителен.

Заметим попутно, что кривые A, B и C располагаются по порядку одна над другой. Это говорит о том, что точность дискертного конечномерного приближения (73) исходной непрерывной задачи (68) уменьшается с ростом номера (модуля) искомого собственного числа.

Кроме того, графики кривых A,B и C имеют минимум. Рост этих кривых вправо и влево от точки минимума обусловлен разными причинами. Так, увеличение погрешности с ростом Δ (в окрестности точки $\log_{10}\Delta = -1$) связано с тем, что шаг дискретизации велик и конечные разности плохо приближают производные. Если же выбрать шаг дискретизации малым (например, $\log_{10}\Delta = -3$), то перестанет выполняться условие на бесконечности (68), что в конечном итоге также ведет к большим погрешностям.

Рассмотренный выше пример убедительно показывает, что использование современных алгоритмов линейной алгебры с гарантированной оценкой точности результата является необходимым для корректного численного исследования физических процессов и открывает новые возможности в установлении степени соответствия между численным экспериментом и реальностью.

Приложение

A Описание программ

Программный пакет состоит из шести модулей: *SingMod, TDiagMod, LSysMod, RefMod, SymMod* и *ArithMod.* (Доступ ко всем модулям обеспечивается оператором *Use AllMod*, который вставляется в начало главной программы.) Первые пять из них являются основными и содержат подпрограммы: решение системы линейных уравнений, обращения матрицы, вычисления сингулярных чисел матрицы, числа обусловленности, собственных значений и векторов симметричных матриц. Последний модуль является служебным. В нём собраны такие процедуры, как определение машинных констант, вычисление уточнённого значения квадратного корня, генерация единичной матрицы, вычисление нормы вектора и другие.

A.1 Модуль *SingMod*

В этом модуле основной является функция *TwoDiagCond*, значение которой есть число обусловленности двухдиагональной матрицы. Так как число обусловленности равно отношению максимального и минимального сингулярных чисел, то очень важной является также процедура *SingValue*, вычисляющая их значения.

A.1.1 Функция NormEstimation

Назначение:

Вычисляет оценку сверху евклидовой нормы квадратной двухдиагональной матрицы D размера N, заданной главной *MainDiag* и побочной *SecondaryDiag* диагоналями. Возвращает вещественное положительное число с двойной точностью, которым оценивается норма двухдиагональной матрицы.

Обращение к функции:

NormEstimation(MainDiag,SecondaryDiag)

Параметры:

MainDiag – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержит элементы главной диагонали двухдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $N-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали двухдиагональной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдаётся сообщение:

Incorrect Arrays Shape in "NormEstimation"

Используемые внешние подпрограммы: отсутствуют.

A.1.2 Подпрограмма SingValue

Назначение:

Вычисляет J – е сингулярное число *Sigma* квадратной двухдиагональной матрицы D размера M, заданной главной *MainDiag* и побочной *SecondaryDiag* диагоналями. Сингулярные числа считаются расположенными по возрастанию. Выдаёт оценку относительной погрешности результата *error*.

Обращение к подпрограмме:

Call SingValue(Sigma,MainDiag,SecondaryDiag,J,error)

Параметры:

Sigma – выходной параметр, действительное число с двойной точностью, содержит вычисленное значение J – го сингулярного числа.

MainDiag – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали квадратной двухдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали квадратной двухдиагональной матрицы.

J – входной параметр, целое положительное число $1 \leq J \leq M$, номер вычисляемого сингулярного числа.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления J – го сингулярного числа:

$$\frac{||Sigma - \sigma_j||}{||D||} \leq error,$$

где σ_j – точное значение j – го сингулярного числа квадратной двухдиагональной матрицы D.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдаётся сообщение:

Incorrect Arrays Shape in "SingValue"

Используемые внешние подпрограммы:
1.*EigenValueThreeDSMatrix (Модуль SymMod)*

A.1.3 Функция TwoDiagCond

Назначение:

Вычисляет число обусловленности квадратной двухдиагональной матрицы D размера N, заданной главной *MainDiag* и побочной *SecondaryDiag* диагоналями. Возвращает вещественное число с двойной точностью равное $\mu = \frac{\sigma_{max}}{\sigma_{min}}$, где σ_{max} , σ_{min} – максимальное и минимальное сингулярные числа исходной матрицы.

Обращение к функции:

TwoDiagCond(MainDiag,SecondaryDiag)

Параметры:

MainDiag – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержит элементы главной диагонали двухдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $N-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали двухдиагональной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдаётся сообщение:

Incorrect Arrays Shape in "TwoDiagCond"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*SingValue (Модуль SingMod)*

A.2 Модуль *TDiagMod*

Подпрограммы этого модуля работают с двухдиагональными матрицами. Процедура *TwoDiagSystemSolution* решает систему с двухдиагональной матрицей коэффициентов, процедура *InvTwoDiagMatrix* производит обращение двухдиагональной матрицы.

A.2.1 Подпрограмма TwoDiagSystemSolution

Назначение:

Программа решает линейную систему алгебраических уравнений с матрицей размера $N \times M$ $(N \geq M)$. Если обозначить за D квадратную верхне-треугольную двухдиагональную матрицу размера $M \times M$

$$D = \begin{pmatrix} d_1 & b_2 & & & \\ & d_2 & b_3 & & \\ & & \ddots & \ddots & \\ & & & d_{M-1} & b_M \\ & & & & d_M \end{pmatrix}, \tag{75}$$

то входная матрица имеет один из трех следующих видов

$$1) \begin{pmatrix} D \\ -- \\ \mathbf{0} \end{pmatrix} \quad \text{при } N > M, \;\; 2)\; D \;\; \text{или} \;\; 3)\; D^* \quad \text{при } N = M. \tag{76}$$

Здесь за $\mathbf{0}$ обозначена нулевая подматрица размера $(N - M) \times M$.

Заданными считаются главная Md и побочная Sd диагонали матрицы D и правая часть RHS (вектор размера N). В качестве решения выдается вектор Sol размера M, а также вектор невязки r размера N и оценка относительной погрешности результата $error$.

Обращение к подпрограмме:

Call TwoDiagSystemSolution(Sol,Md,Sd,RHS,r,error,Position)

Параметры:

Sol – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью. После исполнения программы содержит компоненты вектора решения системы.

Md – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью. Содержит значения элементов главной диагонали квадратной двухдиагональной матрицы D решаемой системы.

Sd – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью. Содержит значения элементов побочной диагонали квадратной двухдиагональной матрицы D решаемой системы.

RHS – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью. Содержит значения компонент вектора правой части системы f.

r – выходной необязательный параметр, одномерный массив длины N вещественных чисел с двойной точностью. Содержит значения компонент вектора невязки системы.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления:

$$\frac{||Sol - x||}{||x||} \leq error,$$

где x – точный вектор решения системы.

Position – входной параметр, целое число, равен нулю в случаях 1) и 2) см. (76) и единице в случае 3). Если параметр *Position* не задан на входе, то считается, что он равен нулю.

Точки остановки и сообщения:
1.Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "TwoDiagSystemSolution"

2. Если обусловленность матрицы D системы $\mu(D) > \frac{\varepsilon_\infty}{2}$ или если норма правой части $||RHS||$ превосходит минимум из $\sigma_1\varepsilon_\infty$ и $\frac{\varepsilon_\infty}{2\mu}$, то есть, если $||RHS|| \geq \min\left\{\sigma_1\varepsilon_\infty\ ,\ \frac{\varepsilon_\infty}{2\mu}\right\}$, где σ_1 – минимальное сингулярное число матрицы системы, μ – обусловленность матрицы системы, ε_∞ – максимальное положительное машинное число, то вычисления могут быть прерваны из-за невозможности размещения результатов промежуточных вычислений в памяти машины при этом выдается одно из следующих предупреждений:

Warning: Calculations can be interrupted, Matrix Condition is too big in "TwoDiagSystemSolution",

Warning: Calculations can be interrupted, Right Hand Side is too big in "TwoDiagSystemSolution".

3. Если норма матрицы D системы слишком велика $||D|| \geq \frac{1}{\varepsilon_0}$ или если норма правой части $||RHS||$ слишком мала

$$||RHS|| \leq \sqrt{M}\varepsilon_0\left[1 + \frac{2}{\varepsilon_1} + \frac{1}{||D||(1-\varepsilon_1)\varepsilon_1}\right],$$

то результат вычисления может быть не точным, поэтому выдается одно из следующих предупреждений:

Warning: Result can be inaccurate, Matrix Norm is too big in "TwoDiagSystemSolution",

Warning: Result can be inaccurate, Right Hand Side is too small in "TwoDiagSystemSolution".

4. В случае, если обусловленность матрицы D системы $\mu(D) > \frac{1}{12\varepsilon_1}$, то оценка ошибки невозможна, о чем свидетельствует предупреждение:

Warning: Estimation is impossible, Matrix Condition is too big in "TwoDiagSystemSolution".

При этом оценка ошибки – параметр *error* принимает значение ε_∞.

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*SingValue (Модуль SingMod)*
3.*VectorNorm (Модуль ArithMod)*

A.2.2 Подпрограмма InvTwoDiagMatrix

Назначение:

Находит обратную матрицу *Inverse* к исходной квадратной двухдиагональной матрице D размера M, заданной главной Md и побочной Sd диагоналями. Выдает оценку относительной погрешности результата error.

Обращение к подпрограмме:

Call InvTwoDiagMatrix(Md,Sd,Inverse,error)

Параметры:

Md – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью. Содержит значения элементов главной диагонали двухдиагональной матрицы .

Sd – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью. Содержит значения элементов побочной диагонали двухдиагональной матрицы .

Inverse – выходной параметр, двумерный массив размера $M \times M$ вещественных чисел с двойной точностью. Содержит значения элементов обратной матрицы.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления обратной матрицы:

$$\frac{\|Inverse - D^{-1}\|}{\|D^{-1}\|} \leq error,$$

где D^{-1} – точное значение обратной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "InvTwoDiagMatrix"

Используемые внешние подпрограммы:
1. *TwoDiagSystemSolution (Модуль TDiagMod)*

A.3 Модуль *LSysMod*

Основные программы этого модуля – *LinSystemSolution* (решение линейной системы) и *InverseMatrix* (обращение квадратной матрицы). Основой использованных в них алгоритмов является приведение матрицы к двухдиагональному виду при помощи ортогональных преобразований. Эти действия осуществляются подпрограммой *TwoDiagonalization* (двухдиагонализация матрицы) с использованием процедур *Reflection* (вычисление преобразования отражения) и *VectorReflection* (применение преобразования отражения к произвольному вектору) из модуля *RefMod*. Подпрограмма *RefinedInverseMatrix* уточняет элементы обратной матрицы итерационным методом.

A.3.1 Подпрограмма TwoDiagonalization

Назначение:

При помощи ортогональных преобразований приводит произвольную матрицу размера $N \times M$, $N \geq M$, к двухдиагональному виду $\begin{pmatrix} D \\ -- \\ \mathbf{0} \end{pmatrix}$, где D – квадратная матрица (75) На входе: $Matrix$ – произвольная матрица размера $N \times M$, $N \geq M$; она же на выходе – результат двухдиагонализации. Также на выходе необязательные параметры: $MainDiag, SecondaryDiag$ – главная и побочная диагонали; $MirLeft$, $MirRight$ – квадратные ортогональные матрицы размеров N и M, приводящие исходную матрицу к двухдиагональному виду. Выдает оценку относительной погрешности двухдиагонализации *error*.

Обращение к подпрограмме:

Call TwoDiagonalization(Matrix,MainDiag,SecondaryDiag,MirLeft,MirRight,error)

Параметры:

Matrix – входной и выходной параметр, двумерный массив размера $N \times M$, $N \geq M$, вещественных чисел с двойной точностью. На входе содержит элементы исходной матрицы, на выходе – вычисленные элементы двухдиагональной матрицы.

MainDiag – выходной необязательный параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержащий вычисленные элементы главной диагонали итоговой квадратной двухдиагональной матрицы.

SecondaryDiag – выходной необязательный параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержащий вычисленные элементы побочной диагонали итоговой квадратной двухдиагональной матрицы.

MirLeft – выходной необязательный параметр, двумерный квадратный массив вещественных чисел с двойной точностью размера N, содержащий произведение преобразований отражения, применяемых слева.

MirRight – выходной необязательный параметр, двумерный квадратный массив вещественных чисел с двойной точностью размера M, содержащий произведение преобразований отражения, применяемых справа.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности двухдиагонализации:

$$\frac{\|B_{\text{маш}} - P_{\text{маш}} A Q^*_{\text{маш}}\|}{\|A\|} \le error,$$

где A – исходная матрица, B – двухдиагональная матрица, полученная в процессе двухдиагонализации, $P_{\text{маш}}$ и $Q^*_{\text{маш}}$ – вычисленные ортогональные преобразования, применяемые соответственно слева и справа.

Точки остановки и сообщения:
1. Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Array Shape in "TwoDiagonalization"

2. Если разнормировка матрицы приводит к переполнению, то работа прекращается с выдачей сообщения:

Warning: Matix and Diagonals Denormalization leads to Overflow in "TwoDiagonalization"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*Eye (Модуль ArithMod)*
3.*Reflection (Модуль RefMod)*
4.*VectorReflection (Модуль RefMod)*

A.3.2 Подпрограмма LinSystemSolution

Назначение:

Применяя процесс двухдиагонализации, подпрограмма решает линейную систему алгебраических уравнений с матрицей $Matrix$ размера $N \times M$ ($N \ge M$ или $N < M$) и правой частью RHS, вектор размера N. В качестве решения выдается вектор Sol размера M, а также вектор невязки r размера N и оценка относительной погрешности результата $error$.

Обращение к подпрограмме:

Call LinSystemSolution(Sol,Matrix,RHS,r,error)

Параметры:

Sol – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержащий компоненты вектора решения системы.

Matrix – входной параметр, двумерный массив размера $N \times M$ вещественных чисел с двойной точностью, матрица коэффициентов линейной системы уравнений.

RHS – входной параметр, одномерный массив длины N, вещественных чисел с двойной точностью, содержащий значения компонент вектора правой части системы.

r – выходной необязательный параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержит значения компонент вектора невязки системы.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности результата:

$$\frac{||Sol - x||}{||x||} \leq error,$$

где x – точный вектор решения системы.

Точки остановки и сообщения:
1. Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "LinSystemSolution"

2. Если число обусловленности матрицы системы $\mu \geq \varepsilon_\infty$, то оценка погрешности результата невозможна, о чем свидетельствует предупреждение:

Warning: Estimation of Result is impossible, Matrix Condition is too big in "LinSystemSolution"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*TwoDiagonalization (Модуль LSysMod)*
3.*TwoDiagSystemSolution (Модуль TDiagMod)*
4.*TwoDiagCond (Модуль SingMod)*

A.3.3 Подпрограмма InverseMatrix

Назначение:

Подпрограмма по исходной квадратной матрице $A = Matrix$ размера N строит обратную к ней *InvMatrix* того же размера. Выдает оценку относительной погрешности результата error.

Обращение к подпрограмме:

Call InverseMatrix(Matrix,InvMatrix,error)

Параметры:

Matrix – входной параметр, двумерный массив размера $N \times N$ вещественных чисел с двойной точностью, содержит элементы исходной матрицы.

InvMatrix – выходной параметр, двумерный массив размера $N \times N$ вещественных чисел с двойной точностью, содержит вычисленные значения обратной матрицы.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности результата:

$$\frac{||InvMatrix - A^{-1}||}{||A^{-1}||} \leq error,$$

где A^{-1} – точная обратная матрица.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "InverseMatrix"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*TwoDiagonalization (Модуль LSysMod)*
3.*InvTwoDiagMatrix (Модуль TDiagMod)*
4.*TwoDiagCond (Модуль SingMod)*

A.3.4 Подпрограмма RefinedInverseMatrix

Назначение:

Для исходной квадратной матрицы $Matrix$ размера M, уточняет элементы ее обратной матрицы $Inverse$, вычисленные, например, подпрограммой $InverseMatrix$. $epsilon$ – заданная точность итерационного процесса. Максимальное число итераций – 50.

Обращение к подпрограмме:

Call RefinedInverseMatrix(Matrix,Inverse,epsilon)

Параметры:

Matrix – входной параметр, двумерный массив размера $M \times M$ вещественных чисел с двойной точностью, содержит элементы исходной матрицы.

Inverse – входной – выходной параметр, двумерный массив размера $M \times M$ вещественных чисел с двойной точностью. На входе содержит значения элементов обратной матрицы, на выходе– уточненные элементы обратной матрицы.

epsilon – входной необязательный параметр, положительное вещественное число с двойной точностью, заданная точность итерационного процесса. Если отсутствует, то $epsilon = 1.0D - 14$.

Точки остановки и сообщения:
1. Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "RefinedInverseMatrix"

2. Если $R = (I - Matrix * Inverse)$ – степень неточности вычисления входной обратной матрицы $Inverse$ больше еденицы, то уточнение невозможно, о чем свидетельствует сообщение:

*Refined is impossible, Norm of R=(I-Matrix*Inverse) is great One in "RefinedInverseMatrix"*

Используемые внешние подпрограммы: отсутствуют.

A.3.5 Функция MatCond

Назначение:

Вычисляет число обусловленности произвольной матрицы $Matrix$ размера $N \times M$, $N \geq M$. Возвращает вещественное число с двойной точностью равное $\mu = \frac{\sigma_{max}}{\sigma_{min}}$, где σ_{max} , σ_{min} – максимальное и минимальное сингулярные числа матрицы.

Обращение к функции:

MatCond(Matrix)

Параметры:

Matrix – входной параметр, двумерный массив размера $N \times M$, $N \geq M$ вещественных чисел с двойной точностью, входная матрица, число обусловленности которой вычисляется.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Array Shape in "MatCond"

Используемые внешние подпрограммы:
1. *TwoDiagonalization (Модуль LSysMod)*
2. *TwoDiagCond(Модуль SingMod)*

A.4 Модуль $RefMod$

Основные программы этого модуля – *Reflection* (вычисление преобразования отражения), *VectorReflection* (применение преобразования отражения к произвольному вектору), которые используются при двухдиагонализации произвольных матриц и трехдиагонализации симметричных матриц.

A.4.1 Подпрограмма Reflection

Назначение:

Если не задан параметр I, программа вычисляет вектор нормали к плоскости, относительно которой должен быть отражен вектор $Vector1$, чтобы быть коллинеарным вектору $Vector2$, и матрицу отражения. Если задан параметр I, то $Vector2$ принимает следующий вид: первые $I-1$ компоненты совпадают с соответствующими компонентами вектора $Vector1$, все компоненты начиная с $I+1$-ой равны нулю, а I – я компонента находится из условия равенства норм $Vector1$ и $Vector2$.

Обращение к подпрограмме:

Call Reflection(Vector1,Vector2,ReflVector,ReflMatrix,I)

Параметры:

Vector1 – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержащий компоненты исходного вектора.

Vector2 – входной – выходной необязательный параметр, одномерный массив длины N, вещественных чисел с двойной точностью, содержащий компоненты вектора, получаемого из исходного в результате действия оператора отражения.

ReflVector – выходной необязательный параметр, одномерный массив длины N, вещественных чисел с двойной точностью, содержащий на выходе вычисленные значения компонент вектора нормали гиперплоскости, относительно которой осуществляется отражение.

ReflMatrix – выходной необязательный параметр, двумерный массив размера $N \times N$, вещественных чисел с двойной точностью, содержащий на выходе вычисленные компоненты матрицы оператора отражения.

I – входной необязательный параметр(обязателен при отсутствии параметра $Vector2$). Целое число $1 \leq I \leq N$. Указывает на то, что искомое преобразование отражения, действуя на вектор $Vector1$, сохраняет первые $I-1$ компоненты, зануляет компоненты начиная с $I+1$ – ой до N – ой, не изменяя при этом нормы вектора.

Точки остановки и сообщения:
1. Если параметры программы не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "Reflection"

2. Если отсутствует параметр I, а элементы массива $Vector2$ слишком малы, то есть

$$\sqrt{\sum_{j=1}^{N} |Vector2(j)|^2} < \varepsilon_0 \ ,$$

где ε_0 – минимальное положительное вещественное число с двойной точностью, то работа прекращается с выдачей соощения:

Vector2 Norm is too small in "Reflection"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*Eye (Модуль ArithMod)*

A.4.2 Подпрограмма VectorReflection

Назначение:

Отражение вектора $Vector$ относительно гиперплоскости с заданным вектором нормали $ReflVector$.

Обращение к подпрограмме:

Call VectorReflection(ReflVector, Vector)

Параметры:

ReflVector – входной параметр, одномерный массив длины N вещественных чисел с двойной точностью, содержащий компоненты вектора нормали к некоторой плоскости.

Vector – входной – выходной, одномерный массив длины N, вещественных чисел с двойной точностью. На входе – содержит компоненты исходного вектора, на выходе – компоненты вектора, получаемого из исходного после его отражения относительно плоскости с нормалью $ReflVector$.

Точки остановки и сообщения:
Если параьетры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "VectorReflection"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*

A.5 Модуль *SymMod*

Подпрограммы этого модуля предназначены для работы с симметричными матрицами. Основой использованных в них алгоритмов является приведение симметричной матрицы к трехдиагональному виду при помощи ортогональных преобразований. Эти действия осуществляются подпрограммой *ThreeDiagonalization* (приведение симметричной матрицы к трехдиагональному виду). Другие главные подпрограммы этого модуля – *EigenValueThreeDSMatrix* (вычисляет собственное значение симметричной трехдиагональной матрицы) и *EigenVectorThreeDSMatrix* (вычисляет нормированный собственный вектор симметричной трехдиагональной матрицы), *EigenValueSymMatrix* (вычисляет собственное значение произвольной симметричной матрицы) и *EigenVectorSymMatrix* (вычисляет нормированный собственный вектор произвольной симметричной матрицы). Функция *NormThreeDiagSMatrix* выдает оценку сверху евклидовой нормы симметричной трехдиагональной матрицы. Значение функции *ThreeDiagCond* – есть число обусловленности симметричной трехдиагональной матрицы.

A.5.1 Подпрограмма ThreeDiagonalization

Назначение:

Приводит произвольную симметричную матрицу $A = SymMatrix$ размера M к трехдиагональному виду при помощи ортогональных преобразований: $S = PAP^*$, где $P = P_1P_2 \dots P_{M-2}$ – произведение преобразований отражения Хаусхолдера.
$SymMatrix$ – входная симметричная матрица, она же на выходе (результат трехдиагонализации) – трехдиагональная симметричная матрица того же размера. Также на выходе необязательные параметры: $MainDiag, SecondaryDiag$ – главная и побочная диагонали трехдиагональной симметричной матрицы, $P = MirLeft$ – квадратная ортогональная матрица размера M, содержащая произведение преобразований отражения Хаусхолдера применяемые слева. Выдает оценку относительной погрешности трехдиагонализации $error$.

Обращение к подпрограмме:

Call ThreeDiagonalization(SymMatrix,MainDiag,SecondaryDiag,MirLeft,error)

Параметры:

SymMatrix – входной и выходной параметр, двумерный квадратный массив размера M, вещественных чисел с двойной точностью. На входе содержит элементы исходной симметричной матрицы, на выходе – вычисленные элементы трехдиагональной симметричной матрицы.

MainDiag – выходной необязательный параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали симметричной трехдиагональной матрицы.

SecondaryDiag – выходной необязательный параметр, одномерный массив длины $M - 1$ – вещественных чисел с двойной точностью, содержит элементы побочной диагонали симметричной трехдиагональной матрицы.

MirLeft – выходной необязательный параметр, двумерный квадратный массив размера M, вещественных чисел с двойной точностью, содержащий произведение преобразований отражения применяемые слева.

error – выходной необязательный параметр, положительное вещественное число с двойной точностью, содержит оценку относительной погрешности трехдиагонализации матрицы:

$$\frac{||S - PAP^*||}{||A||} \leq error,$$

где A – исходная симметричная матрица, S – трехдиагональная симметричная матрица, полученная в результате процесса трехдиагонализации.

Точки остановки и сообщения:
1. Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "ThreeDiagonalization"

2. Если размеры матрицы слишком велики, т.е. неравенство (11) не выполнено, то оценка ошибки трехдиагонализации может быть не точной. Программа предупреждает об этом сообщением:

Warring: Estimatin can be inaccurate, Matrix Size is too big in "ThreeDiagonalization"

3. Если разнормировка матрицы приводит к переполнению, то работа прекращается с выдачей сообщения:

Warning: Matix and Diagonals Denormalization leads to Overflow in "ThreeDiagonalization"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*Eye (Модуль ArithMod)*
3.*Reflection (Модуль RefMod)*
4.*VectorReflection (Модуль RefMod)*
5.*NormThreeDSMatrix (Модуль SymMod)*

A.5.2 Подпрограмма EigenValueThreeDSMatrix

Назначение:

Вычисляет, методом бисекций, J – е собственное значение *Lambda* симметричной трехдиагональной матрицы S размера M ,заданной главной *MainDiag* и побочной *SecondaryDiag* диагоналями. Собственные значения считаются расположенными по возрастанию. Выдает оценку относительной погрешности результата *error*.

Обращение к подпрограмме:

Call EigenValueThreeDSMatrix(MainDiag,SecondaryDiag,J,Lambda,error)

Параметры:

MainDiag – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали симметричной трехдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали симметричной трехдиагональной матрицы.

J – входной параметр, целое положительное число $1 \leq J \leq M$, номер вычисляемого собственного значения.

Lambda – выходной параметр, действительное число с двойной точностью, содержит вычисленное значение J – го собственного числа.

error – выходной необязательный параметр, вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления *Lambda*:

$$\frac{||Lambda - \lambda_j||}{||S||} \leq error,$$

где λ_j – точное j – ое собственное значение исходной симметричной трехдиагональной матрицы S.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "EigenValueThreeDSMatrix"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*NormalizationMatrix (Модуль SymMod)*
3.*KShturm (Модуль SymMod)*
4.*NormThreeDSMatrix (Модуль SymMod)*

A.5.3 Подпрограмма EigenVectorThreeDSMatrix

Назначение:

Для заданного n – того собственного значения $Lambda$ симметричной трехдиагональной матрицы S размера M, с относительной погрешностью $errorLambda$, вычисляет компоненты нормированного собственного вектора $EigenVector$ через элементы двусторонней последовательности Штурма данной матрицы, заданной главной $MainDiag$ и побочной $SecondaryDiag$ диагоналями. Выдает оценку относительной погрешности результата $error$.

Обращение к подпрограмме:

Call EigenVectorThreeDSMatrix(EigenVector,MainDiag,SecondaryDiag,n,Lambda,errorLambda,error)

Параметры:

Lambda – входной параметр, действительное число с двойной точностью, содержит приближенное значение n – го собственного числа входной симметричной трехдиагональной матрицы.

errorLambda – входной параметр, вещественное число с двойной точностью, содержит оценку относительной ошибки *Lambda*.

n – входной параметр, целое положительное число $1 \leq n \leq M$, номер n–го собственного значения входной симметричной трехдиагональной матрицы.

MainDiag – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали входной симметричной трехдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали входной симметричной трехдиагональной матрицы.

EigenVector – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит компоненты нормированного собственного вектора входной симметричной трехдиагональной матрицы, соответствующего n – му собственному значению λ_n.

error – выходной необязательный параметр, вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления $EigenVector$.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "EigenVectorThreeDSMatrix"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*NormalizationMatrix (Модуль SymMod)*
3.*NormThreeDSMatrix (Модуль SymMod)*

A.5.4 Подпрограмма EigenValueSymMatrix

Назначение:

Вычисляет J – е собственное значение $Lambda$ произвольной симметричной матрицы $A = SymMatrix$ размера M, равное J – му собственному значению трехдиагональной симметричной матрицы полученной в результате трехдиагонализации исходной. Собственные значения считаются расположенными по возрастанию. Выдает оценку относительной погрешности результата $error$.

Обращение к подпрограмме:

Call EigenValueSymMatrix(SymMatrix,J,Lambda,error)

Параметры:

SymMatrix – входной параметр, квадратный двухмерный массив размера M вещественных чисел с двойной точностью, содержит элементы исходной симметричной матрицы.

J – входной параметр, целое положительное число $1 \leq J \leq M$, номер вычисляемого J – го собственного значения матрицы.

Lambda – выходной параметр, действительное число с двойной точностью, содержит вычисленное значение J-го собственного числа входной симметричной матрицы.

error – выходной необязательный параметр, вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления $Lambda$:

$$\frac{||Lambda - \lambda_j||}{||A||} \leq error,$$

где λ_j – точное j – ое собственное значение исходной симметричной матрицы A.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "EigenValueSymMatrix"

Используемые внешние подпрограммы:
1.*ThreeDiagonalization(Модуль SymMod)*
2.*EigenValueThreeDSMatrix(Модуль SymMod)*

A.5.5 Подпрограмма EigenVectorSymMatrix

Назначение:

Для заданного J – го собственного значения $Lambda$ произвольной симметричной матрицы $SymMatrix$ размера M, с относительной погрешностью $errorLambda$, вычисляет компоненты нормированного собственного вектора. Для этого входная симметричная матрица ортогональными преобразованиями приводится к трехдиагональному виду. Определяются компоненты нормированного собственного вектора симметричной трехдиагональной матрицы через элементы двусторонней последовательности Штурма матрицы. Собственный вектор $EigenVector$ исходной матрицы равен произведению транспонированной матрицы ортогонального преобразования и собственного вектора симметричной трехдиагональной матрицы. Выдает оценку относительной погрешности результата $error$.

Обращение к подпрограмме:

Call EigenVectorSymMatrix(EigenVector,SymMatrix,J,Lambda,errorLambda,error)

Параметры:

SymMatrix – входной параметр, квадратный двухмерный массив размера M вещественных чисел с двойной точностью, содержит элементы исходной симметричной матрицы.

J – входной параметр, целое положительное число $1 \leq J \leq M$, заданный номер J – го собственного значения исходной симметричной матрицы.

Lambda – входной параметр, действительное число с двойной точностью, содержит значение J – го собственного числа.

errorLambda – входной параметр, вещественное число с двойной точностью, содержит оценку относительной ошибки *Lambda*.

EigenVector – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит компоненты нормированного собственного вектора входной симметричной матрицы, соответствующего j – му собственному значению λ_j.

error – выходной необязательный параметр, вещественное число с двойной точностью, содержит оценку относительной погрешности вычисления $EigenVector$.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "EigenVectorSymMatrix"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*ThreeDiagonalization(Модуль SymMod)*
3.*EigenValueThreeDSMatrix(Модуль SymMod)*
4.*EigenVectorThreeDSMatrix(Модуль SymMod)*

A.5.6 Функция SymMatCond

Назначение:

Вычисляет число обусловленности произвольной симметричной матрицы $SymMatrix$ размера M (отношение максимального и минимального сингулярных чисел матрицы). Сингулярные числа симметричной матрицы равны абсолютным значениям собственных чисел этой матрицы. Возвращает вещественное число с двойной точностью, равное $\mu = \frac{\sigma_{max}}{\sigma_{min}}$.

Обращение к функции:

SymMatCond(SymMatrix)

Параметры:

SymMatrix – входной параметр, квадратный двухмерный массив размера M вещественных чисел с двойной точностью, содержит элементы исходной симметричной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Array Shape in "SymMatCond"

Используемые внешние подпрограммы:
1.*ThreeDiagonalization (Модуль SymMod)*
2.*ThreeDiagCond (Модуль SymMod)*

A.5.7 Функция ThreeDiagCond

Назначение:

Вычисляет число обусловленности трехдиагональной симметричной матрицы размера M, заданной главной $MainDiag$ и побочной $SecondaryDiag$ диагоналями. Возвращает вещественное число с двойной точностью равное $\mu = \frac{\sigma_{max}}{\sigma_{min}}$, где σ_{max} , σ_{min} – максимальное и минимальное сингулярные числа исходной матрицы, которые для симметричной матрицы равны ее максимальному и минимальному по модулю собственным числам.

Обращение к функции:

ThreeDiagCond(MainDiag,SecondaryDiag)

Параметры:

MainDiag – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали симметричной трехдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали симметричной трехдиагональной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "ThreeDiagCond"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*
2.*EigenValueThreeDSMatrix (Модуль SymMod)*
3.*NormalizationMatrix (Модуль SymMod)*
4.*KShturm (Модуль SymMod)*

A.5.8 Функция NormThreeDSMatrix

Назначение:

Вычисляет оценку сверху евклидовой нормы симметричной трехдиагональной матрицы размера M, заданной главной *MainDiag* и побочной *SecondaryDiag* диагоналями. Возвращает вещественное положительное число с двойной точностью, которым оценивается норма трехдиагональной матрицы.

Обращение к функции:

NormThreeDSMatrix(MainDiag,SecondaryDiag)

Параметры:

MainDiag – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали симметричной трехдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали симметричной трехдиагональной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "NormThreeDSMatrix"

Используемые внешние подпрограммы: отсутствуют.

A.5.9 Подпрограмма NormalizationMatrix

Назначение:

Проводит специальную подготовку исходной симметричной трехдиагональной матрицы, заданной диагоналями $MainDiag$ и $SecondaryDiag$, необходимую для безаварийного вычисления последовательностей Штурма. Сюда входит нормировка матрицы (умножение матрицы на число ro, подобранное так, чтобы максимальный из ненулевых элементов матрицы лежал в определенных границах) и замена всех нулевых элементов на главной и побочных диагоналях на специальное число. Возвращает нормировочный множитель ro – вещественное число с двойной точностью, равное степени основания системы счисления γ^k (γ для конкретной ЭВМ определяется подпрограммой *MachConst*) и нормированные диагонали d и b матрицы.

Обращение к подпрограмме:

Call NormalizationMatrix(MainDiag,SecondaryDiag,ro,d,b)

Параметры:

MainDiag – входной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит элементы главной диагонали симметричной трехдиагональной матрицы.

SecondaryDiag – входной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит элементы побочной диагонали симметричной трехдиагональной матрицы.

ro – выходной параметр, вещественное число с двойной точностью, нормировочный множитель.

d – выходной параметр, одномерный массив длины M вещественных чисел с двойной точностью, содержит нормированные элементы главной диагонали симметричной трехдиагональной матрицы.

b – выходной параметр, одномерный массив длины $M-1$ вещественных чисел с двойной точностью, содержит нормированные элементы побочной диагонали симметричной трехдиагональной матрицы.

Точки остановки и сообщения:
Если параметры не соответствуют описанию, программа прекращает работу. При этом выдается сообщение:

Incorrect Arrays Shape in "NormalizationMatrix"

Используемые внешние подпрограммы:
1.*MachConst (Модуль ArithMod)*

A.6 Модуль *ArithMod*

Служебный модуль, в котором собраны такие процедуры, как определение машинных констант, вычисление уточненного значения квадратного корня, генерация единичной матрицы, вычисление нормы вектора и матрицы.

A.6.1 Подпрограмма MachConst

Назначение: определяет машиные константы
$\gamma = Gamma$, $\varepsilon_0 = Eps0$, $\varepsilon_1 = Eps1$, $\varepsilon_\infty = EpsMax$.
Обращение к подпрограмме: *Call MachConst*

A.6.2 Функция SqRoot

Назначение: специальное извлечение квадратного корня из неотрицательного машинного числа гарантирующее относительную погрешность результата.
Обращение к функции: *SqRoot(x)*
Параметр: входной параметр – вещественное число с двойной точностью.
Результат: вещественное число с двойной точностью, уточненное значение квадратного корня.

A.6.3 Подпрограмма Eye

Назначение: входную квадратную матрицу заменяет на единичную.
Обращение к подпрограмме: *Call Eye(A)*
Параметр: A – входной – выходной параметр, двумерный массив вещественных чисел с двойной точностью размера $N \times N$. На выходе содержит единичную матрицу.
Точки остановки и сообщения:
Если массив A не квадратный, программа прекращает работу с сообщением:

Incorrect Array Shape in "Eye".

A.6.4 Функция VectorNorm

Назначение: вычисляет норму вектора, используя функцию *SqRoot*.
Обращение к функции: *VectorNorm(Vector)*
Параметр: *Vector* – входной параметр, одномерный массив вещественных чисел с двойной точностью длины N.
Результат: вещественное число с двойной точностью, вычисленная норма вектора.

A.6.5 Функция FrobeniusNormMatrix

Назначение: вычисляет фробениусову норму матрицы.
Обращение к функции: *FrobeniusNormMatrix(Matrix)*
Параметр: *Matrix* – входной параметр, двумерный массив вещественных чисел с двойной точностью размера $N \times M$.
Результат: вещественное число с двойной точностью, вычисленная норма матрицы.

Литература

[1] Ф.Р. Гантмахер, *Теория матриц*, Наука, Москва (1967).

[2] С.К. Годунов, А.Г. Антонов, О.П. Кирилюк,В.И. Костин, *Гарантированная точность решения систем линейных уравнений в евклидовых пространствах*, Наука, Новосибирск (1992).

[3] Э.Ф. Бибердорф, Н.И. Попова, *Решение линейных систем с гарантированной оценкой точности результатов (часть первая)* Препринт ИЯФ СО РАН 99-49, Новосибирск,(1999).

[4] С.К. Годунов, В.С. Рябенький *Введение в теорию разностных схем*, Москва, ГИФМЛ (1962).

[5] А.С. Хаусхольдер, *Основы численного анализа*, Москва, ИЛ (1956).

[6] А.Н. Малышев, *Введение в вычислительную линейную алгебру*, Наука, Новосибирск (1991).

[7] Дж.Х. Уилкинсон, *Алгебраическая проблема собственных значений*, Наука, Москва (1970).

[8] Д.К. Фаддеев, В.Н. Фаддеева, *Вычислительные методы линейной алгебры*, Физматгиз, Москва (1960).

[9] D.H. Bailey, *ACM Translation on Math. Software* 19, 288 (1993).

[10] А.Ф. Никифоров, В.Б. Уваров, *Основы теории специальных функций*, Наука, Москва (1974).