

УДК 512.1; 517.53; 519.6

Ю.А. Несмеев

АЛГОРИТМ РЕШЕНИЯ КУБИЧЕСКОГО УРАВНЕНИЯ

Предложен алгоритм нахождения корней кубического уравнения. Он использует метод неопределённых коэффициентов и дихотомию. Он учитывает: взаимное расположение параболы и гиперболы на плоскости; расположение корней алгебраического уравнения на комплексной плоскости. Предложена модификация алгоритма. Даны примеры применения и алгоритма, и его модификации. В одном из примеров вычисляются посредством алгоритма корни полинома Якоби.

Ключевые слова: *дихотомия, комплексная плоскость, корень, уравнение.*

В настоящее время в справочной литературе для научно-технических работников практически только в [1, с. 29] представлен численный способ решения кубического уравнения. Способ из [1] заключается в последовательном выполнении следующих действий: α) поиск (путём нахождения значений левой части уравнения в отдельных точках числовой оси) того замкнутого промежутка, который содержит во внутренней точке один корень, имеющий кратностью число 1; β) вычисление начального приближения этого корня обратной линейной интерполяцией; γ) итерационное вычисление следующих приближений этого корня по формуле метода касательных [2, с. 421]; δ) деление левой части уравнения на разность между неизвестной величиной и последним приближением этого корня; ϵ) вычисление корней квадратного многочлена, полученного предыдущим действием. Этот способ на практике часто требует больших затрат времени. Притом наибольшие затраты приходится на действия α , β и γ , то есть на использование метода касательных. Значительных затрат требует и действие δ (при воплощении в компьютерные программы). Поэтому целью исследований, представленных в данной работе, было построение такого доступного для научно-технических работников численного способа решения уравнения

$$x^3 + k_1x^2 + k_2x + k_3 = 0 \quad (1)$$

с действительными коэффициентами, который не использует метод касательных. Задачей исследования служило построение такого алгоритма вычисления корней уравнения (1), который не выполняет действий α , β , γ , δ и использует дихотомию. Задача исследования была решена в предположении $k_3 \neq 0$. Методами исследования были изучение и анализ литературных источников, теоретически обоснованные математические выкладки, математическое моделирование и компьютерный эксперимент. Решение задачи изложено ниже.

Применение метода неопределённых коэффициентов к преобразованию левой части уравнения (1) приводит к тождеству

$$x^3 + k_1x^2 + k_2x + k_3 \equiv (x^2 + a^*x + b^*)(x + c^*), \quad (2)$$

в котором совокупность величин a^* , b^* , c^* является решением системы

$$a + c = k_1, \quad (3.1)$$

$$ac + b = k_2, \quad (3.2)$$

$$bc = k_3. \quad (3.3)$$

Решение a^* , b^* , c^* может быть неоднозначным, но для нахождения корней уравнения (1) достаточно найти одно решение системы (3). Ниже этот факт учитывается при построении алгоритма. Алгоритм использует в формулировках шагов 4 и 6 следующие обозначения корней уравнения (1): тот действительный корень, который получается приравниванием к нулю выражения $x + c^*$, имеет обозначение x_1 ; те корни, которые получаются приравниванием к нулю квадратного трехчлена $x^2 + a^*x + b^*$, имеют обозначения x_2 и x_3 . Согласно принятым обозначениям, справедливо равенство $x_1 = -c^*$, которое с помощью зависимости (3.1) преобразуется в соотношение $x_1 = a^* - k_1$. Это соотношение используется в формулировке шага 4 алгоритма. Тот факт, что x_2 и x_3 являются корнями трёхчлена $x^2 + a^*x + b^*$, используется в формулировке шага 6 алгоритма.

Система (3) равносильна системе

$$c = k_1 - a, \quad (4.1)$$

$$a(k_1 - a) + b = k_2, \quad (4.2)$$

$$b(k_1 - a) = k_3. \quad (4.3)$$

Так как в уравнениях (4.2) и (4.3) отсутствует величина c , то лишь с помощью этих уравнений можно найти a^* и b^* . Используем эту возможность, отнеся плоскость к прямоугольным декартовым координатам a, b с центром в точке O . Будем считать, что плоскость изображена на экране монитора (заполняя его полностью), а точка O изображается его центром. Также будем считать, что относительно взгляда пользователя компьютером на экран координатная ось a направлена вправо от центра, а координатная ось b – вверх от центра. Преобразуем соотношения (4.2) и (4.3):

$$b = a^2 - k_1a + k_2, \quad (5.1)$$

$$b = -k_3 / (a - k_1). \quad (5.2)$$

Из соотношения (5.1) следует, что $b^* = a^*(a^* - k_1) + k_2$. Этот факт используется в формулировке шага 5 алгоритма.

Зависимость (5.1) задаёт на плоскости параболу, ветви которой направлены вверх. Ось симметрии параболы в случае $k_1 = 0$ совпадает с координатной осью b , а в случае $k_1 \neq 0$ параллельна ей. Абсциссой и ординатой вершины параболы являются соответственно числа $2^{-1} \cdot k_1$ и $k_2 - 4^{-1} \cdot k_1^2$. Зависимость (5.2) определяет на плоскости гиперболу, центром которой служит та точка, абсциссой и ординатой которой являются соответственно числа k_1 и 0. Асимптотами гиперболы являются координатная ось a и прямая $a = k_1$. Вертикальная асимптота гиперболы пересекается с параболой в той точке, абсциссой и ординатой которой являются соответственно числа k_1 и k_2 .

Исключение величины b из уравнений (4.2) и (4.3) приводит к следующему уравнению для нахождения абсциссы a^* точки пересечения гиперболы и параболы:

$$a^3 - 2k_1a^2 + (k_1^2 + k_2)a - k_1k_2 + k_3 = 0. \quad (6)$$

Заменим в нём обозначение неизвестной величины символом t , а вместо a^* будем использовать обозначение t^* . Построим (на основе левой части уравнения (6)) вспомогательный многочлен q , определяемый равенством

$$q(t) = t^3 - 2k_1t^2 + (k_1^2 + k_2)t - k_1k_2 + k_3. \quad (7)$$

Рассмотрим на комплексной z -плоскости соответствующее многочлену q уравнение

$$z^3 - 2k_1z^2 + (k_1^2 + k_2)z - k_1k_2 + k_3 = 0, \quad (8)$$

в котором $z = a + ib$. Воспользуемся (с точностью до обозначений) известными свойствами корней алгебраического уравнения с действительными коэффициентами на комплексной плоскости [3, с. 75]. Согласно этим свойствам, корни уравнения (8) на z -плоскости принадлежат замкнутому кольцу или замкнутому кругу. Центром кольца или круга служит начало координат. В случае кольца границей является объединение двух окружностей, имеющих свои радиусы. В случае круга границей служит его окружность. В данной работе в случае кольца меньший радиус имеет обозначение n_b , а больший радиус – n_c . В случае круга за n_b принято число 0, а за n_c – радиус круга. Величины n_b и n_c зависят от величин b_3 и c_0 , определяемых равенствами

$$b_3 = \max\{1, |-2k_1|, |k_1^2 + k_2|\}; \quad (9)$$

$$c_0 = \max\{|-2k_1|, |k_1^2 + k_2|, |-k_1k_2 + k_3|\}. \quad (10)$$

Для величин n_b и n_c справедливы зависимости

$$n_b = |-k_1k_2 + k_3|/(b_3 + |-k_1k_2 + k_3|), n_c = 1 + c_0. \quad (11)$$

Зависимости (9) – (11) учитываются на шаге 1 алгоритма.

Величина t^* , являясь корнем и многочлена (7), и уравнения (8), располагается на действительной оси комплексной z -плоскости, а именно в объединении замкнутых промежутков

$$[-n_c, -n_b], [n_b, n_c]. \quad (12)$$

Из величин n_b и n_c лишь n_b может принимать значение 0. Этот факт учитывается на шаге 2.

Если уравнение (8) имеет комплексные корни, а его действительный корень отличен от нуля, то этот корень является внутренней точкой одного из промежутков (12). Поэтому он может быть найден с помощью дихотомии одного из промежутков (12) и присваивания величине t^* значения этого корня. Если уравнение (8) не имеет комплексных корней и ни один из его корней не является нулём, то при любом расположении действительных корней уравнения (8) внутри промежутков (12) один из них позволяет найти с помощью дихотомии корень уравнения (8) и присвоить величине t^* значения найденного корня. Эти факты являются очевидными и используются на шаге 3 алгоритма.

Изложенное выше позволяет сформулировать алгоритм решения уравнения (1).

Алгоритм

1. Вычисление значений величин b_3 , c_0 , n_b , n_c по формулам

$$b_3 = \max\{1, |-2k_1|, |k_1^2 + k_2|\},$$

$$c_0 = \max\{|-2k_1|, |k_1^2 + k_2|, |-k_1k_2 + k_3|\},$$

$$n_b = |-k_1k_2 + k_3|/(b_3 + |-k_1k_2 + k_3|), n_c = 1 + c_0.$$

2. Проверка величин n_b на корень уравнения $z^3 - 2k_1z^2 + (k_1^2 + k_2)z - k_1k_2 + k_3 = 0$. В случае положительного результата проверки: присваивание величине t^* значения n_b и переход на шаг 4.

3. Определение знака произведения $q(n_b) \cdot q(n_c)$. В случае отрицательного знака – дихотомия промежутка $[n_b, n_c]$ и присваивание величине t^* значения найденного корня. В случае положительного знака – дихотомия промежутка $[-n_c, -n_b]$ и присваивание величине t^* значения найденного корня.

4. Вычисление действительного корня x_1 уравнения (1) по формуле $x_1 = t^* - k_1$.

5. Вычисление значений величин a^* и b^* по формулам

$$a^* = t^*, b^* = a^*(a^* - k_1) + k_2.$$

6. Вычисление корней уравнения $x^2 + a^*x + b^* = 0$ и присваивание их значений корням x_2 и x_3 уравнения (1).

Пример 1. Требуется решить с помощью алгоритма уравнение $x^3 - 18,1x - 34,8 = 0$ [1, с. 29]. Перед применением алгоритма находим значения величин k_1, k_2, k_3 : $k_1 = 0$; $k_2 = -18,1$; $k_3 = -34,8$.

Результаты выполнения шагов алгоритма:

1) $b_3 = 18,1$; $c_0 = 34,8$; $n_b = 0,657844990548204$; $n_c = 35,800000000000000$.

2) $q(n_b) \neq 0$.

3) $q(n_b) \cdot q(n_c) < 0$. $t^* = 5,005265097281269$.

4) $x_1 = 5,005265097281269$.

5) $a^* = 5,005265097281269$; $b^* = 6,952678694062071$.

6) $x_2 = -2,502632548640635 + i(-0,830366798798310)$;

$x_3 = -2,502632548640635 + i(0,830366798798310)$.

При решении уравнения из примера 1 посредством алгоритма итерационные шаги были прекращены тогда, когда абсолютная величина значения многочлена q в середине очередного отрезка не превысила числа 10^{-17} . Всего было сделано 65 шагов. Корни, полученные с помощью алгоритма, совпали с корнями, полученными по формулам из [4].

Сопоставление корней, полученных с помощью алгоритма, с корнями из [1] приводит к следующим выводам. При округлении x_1 и действительных частей корней x_2 и x_3 , полученных с помощью алгоритма, соответственно до шести и девяти значащих цифр получаются соответствующие величины из [1]. При таком округлении мнимых частей корней x_2 и x_3 , полученных с помощью алгоритма, при котором число значащих цифр совпадает с числом значащих цифр в корнях x_2 и x_3 из [1], имеет место несовпадение в трёх последних разрядах.

Пример 2. Требуется вычислить корни полинома Якоби $P_3(2,1,x)$ [5, с. v] с помощью алгоритма. Для вычисления корней предварительно найдём значения коэффициентов k_1, k_2, k_3 по следствию из формулы явного выражения [5, с. v]. В результате получим следующие равенства:

$$k_1 = -1,28571428571429; k_2 = 4,28571428571429 \cdot 10^{-1};$$

$$k_3 = -2,85714285714286 \cdot 10^{-2}.$$

Результаты выполнения шагов алгоритма:

1) $b_3 = 2,57142857142857$, $c_0 = 2,57142857142857$, $n_b = 1,68865435356201 \cdot 10^{-1}$, $n_c = 3,57142857142857$.

2) $q(n_b) \neq 0$.

3) $q(-n_c) \cdot q(-n_b) < 0$, $t^* = -1,19712632620158$.

4) $x_1 = 8,85879595127039 \cdot 10^{-2}$.

5) $a^* = -1,1971263262015$, $b^* = 3,22520450054291 \cdot 10^{-1}$.

6) $x_2 = 7,87659461760847 \cdot 10^{-1}$, $x_3 = 4,09466864440735 \cdot 10^{-1}$.

При решении уравнения из примера 2 посредством алгоритма итерационные шаги были прекращены тогда, когда абсолютная величина многочлена q в сере-

дине очередного отрезка не превысила число 10^{-17} . Всего было сделано 55 шагов. Все корни, полученные с помощью алгоритма, совпали с корнями, полученными по формулам из [4]. Корни x_1 и x_2 , полученные с помощью алгоритма, совпадают с соответствующими известными корнями [5, с. 235] (в [5] приведены корни, полученные методом Ньютона). Притом последняя значащая цифра корня x_3 , полученного с помощью алгоритма, отличается на единицу от соответствующей цифры корня из [5], а остальные значащие цифры полученного с помощью алгоритма корня x_3 совпадают с соответствующими цифрами корня из [5].

Значения величин a^* и b^* в примерах 1 и 2 являются координатами точек, расположенных в верхней полуплоскости той плоскости, которая выше отнесена к системе координат O, a, b . В обоих примерах значение величины a^* больше значения величины k_1 , а величина k_3 имеет отрицательное значение. Эти факты – проявление следующих закономерностей: если $k_3 < 0$, то в верхней полуплоскости, справа от вертикальной асимптоты гиперболы, парабола и гипербола имеют по крайней мере одну общую точку; если $k_3 > 0$, то в верхней полуплоскости, слева от вертикальной асимптоты гиперболы, парабола и гипербола имеют по крайней мере одну общую точку. Закономерности вытекают из взаимного расположения гиперболы и параболы на плоскости. Алгоритм закономерностей не учитывает, хотя учёт может приводить к уменьшению длины того промежутка, к которому можно применять дихотомию. Учёт закономерностей приводит к изложенной ниже модификации алгоритма. Модификация использует табл. 1, в которой тот промежуток, к которому следует применять дихотомию, имеет обозначение W . Если одной из его границ является k_1 , то имеет место упоминаемое выше уменьшение длины промежутка, подвергаемого дихотомии.

Таблица 1

Границы предназначенного для дихотомии промежутка W

Случай	Меньшая и большая границы промежутка W	
1	$k_3 < 0; k_1 \geq 0; q(k_1) \cdot q(n_c) < 0$	k_1 n_c
2	$k_3 < 0; k_1 < 0; q(k_1) \cdot q(-n_b) < 0$	k_1 $-n_b$
3	$k_3 < 0; k_1 < 0; q(n_b) \cdot q(n_c) < 0$	n_b n_c
4	$k_3 > 0; k_1 \leq 0; q(-n_c) \cdot q(k_1) < 0$	$-n_c$ k_1
5	$k_3 > 0; k_1 > 0; q(-n_c) \cdot q(-n_b) < 0$	$-n_c$ $-n_b$
6	$k_3 > 0; k_1 > 0; q(n_b) \cdot q(k_1) < 0$	n_b k_1

Модификация алгоритма

1. Вычисление значений величин b_3, c_0, n_b, n_c по формулам

$$b_3 = \max\{1, |-2k_1|, |k_1^2 + k_2|\},$$

$$c_0 = \max\{|-2k_1|, |k_1^2 + k_2|, |-k_1k_2 + k_3|\},$$

$$n_b = |-k_1k_2 + k_3| / (b_3 + |-k_1k_2 + k_3|), \quad n_c = 1 + c_0.$$

2. Установление номера случая по табл. 1.

3. Проверка величины n_b на корень уравнения $z^3 - 2k_1z^2 + (k_1^2 + k_2)z - k_1k_2 + k_3 = 0$. При положительном результате проверки: идентификация величины n_b как t^* и переход на шаг 4. При отрицательном результате проверки: дихотомия (в целях поиска корня многочлена q) промежутка W и идентификация найденного корня как величины t^* .

4. Вычисление действительного корня x_1 уравнения (1) по формуле $x_1 = t^* - k_1$.

5. Вычисление значений величин a^* и b^* по формулам

$$a^* = t^*, b^* = a^*(a^* - k_1) + k_2.$$

6. Вычисление корней уравнения $x^2 + a^*x + b^* = 0$ и идентификация их как корней x_2 и x_3 уравнения (1).

Пример 3. Требуется решить с помощью модификации алгоритма уравнение $x^3 + \pi x + 100 = 0$. Перед применением алгоритма находим значения величин k_1, k_2, k_3 : $k_1 = 0; k_2 = \pi; k_3 = 100$.

Результаты выполнения шагов
модификации алгоритма:

- 1) $b_3 = 3,14159265358979, c_0 = 1,00000000000000 \cdot 10^2,$
 $n_b = 9,69540972048579 \cdot 10^{-1}, n_c = 1,01000000000000 \cdot 10^2.$
- 2) Случай 4.
- 3) $t^* = -4,41616324489854.$
- 4) $x_1 = -4,41616324489854.$
- 5) $a^* = -4,41616324489854, b^* = 2,26440904591826 \cdot 10^1.$
- 6) $x_2 = 2,20808162244927 + i \cdot 4,21526582883979.$
 $x_3 = 2,20808162244927 - i \cdot 4,21526582883979.$

При решении уравнения из примера 3 деление пополам очередного отрезка было прекращено тогда, когда его длина не превысила числа 10^{-18} . Всего было сделано 67 итерационных шагов. Все корни, полученные с помощью модификации алгоритма, совпали с корнями, полученными по формулам из [4].

При получении вышеприведённых результатов, связанных с примерами 1–3, использовались специально разработанные автором компьютерные программы на языке Турбо Паскаль. В них для арифметических операций над числами и операции извлечения квадратного корня из неотрицательного числа применялись соответственно арифметические выражения [6, с. 146] и функция *sqrt* [6, с. 208] из стандарта языка программирования. Притом в программах все величины, имеющие дробные значения, имели тип *extended*, благодаря чему представления этих величин в компьютере содержало 19–20 цифр.

Для испытания предлагаемого алгоритма на возможность его применения к высокоточному вычислению (постановка вопроса принадлежит Е. М. Малекю из Магнитогорского государственного технического университета) действительный корень x_1 и компоненты корней x_2 и x_3 уравнения из примера 3 также были вычислены с точностью 50 знаков после запятой. Притом применялось моделирование шагов алгоритма, а в его рамках, в частности, и моделирование арифметических операций над числами и моделирование операции извлечения квадратного корня из неотрицательного числа. (При моделировании не использовались арифметические выражения и функция извлечения квадратного корня, содержащиеся в стандарте языка программирования.) Вычисление было произведено в среде языка Турбо Паскаль и описано ниже.

При моделировании выполнялись действия над строками. Каждая строка содержала 60 символов. Это число обеспечило, в частности, сходимость итерационных процессов, имитирующих при моделировании вычисление корня методом дихотомии и итерационного извлечения квадратного корня из неотрицательного числа. Первым символом строки был символ *пробел* в случае неотрицательного числа и символ *минус* в случае отрицательного числа. Те строки, которые заменя-

Таблица 3

Результаты моделирования шагов алгоритма

Шаг	Результаты моделирования шага
1	'b ₃ ' = ' 3,141592653589793238462643383279502884197169399375105820974'; 'c ₀ ' = ' 100,00'; 'n _b ' = ' 0,969540972048578778409308508755311638040661219809103898400'; 'n _c ' = ' 101,00'.
2	'q(n _b)' ≠ ' 0,00 '.
3	Первым символом строки 'q(-n _c)-q(-n _b)' является знак <i>минус</i> ; 't*' = '-4,416163244898541926876995816539042620956252871850369939347'.
4	'x ₁ ' = '-4,416163244898541926876995816539042620956252871850369939347'.
5	'a*' = '-4,416163244898541926876995816539042620956252871850369939347'; 'b*' = ' 22,64409045918261243660427024591830200837301521253241874573'.
6	'rex ₂ ' = ' 2,208081622449270963438497908269521310478126435925184969673'; 'imx ₂ ' = ' 4,215265828839790696631198927569780758976854985761136033563'.

Округление чисел, соответствующих строкам 'x₁', rex₂, 'imx₂' из табл. 3, до пятидесяти знаков после запятой приводит к тем расположенным в табл. 4 значениям корней, которые соответствуют условиям примера при вычислении (действительного корня и компонентов комплексных корней с ненулевыми мнимыми частями) с точностью 50 знаков после запятой.

Таблица 4

Полученные моделированием корни

Обозначение	Корень
x ₁	- 4,41616324489854192687699581653904262095625287185037
x ₂	2,20808162244927096343849790826952131047812643592518 + + i4,215 26582883979069663119892756978075897685498576114
x ₃	2,20808162244927096343849790826952131047812643592518 - - i4,21526582883979069663119892756978075897685498576114

Значение округлённого до пятидесяти знаков после запятой корня x₁, полученного моделированием шагов алгоритма, совпало с результатом вычислений Е.М. Малек по методу Ньютона с помощью математического пакета Maple 8. Значения округлённых до пятидесяти знаков после запятой компонентов корней x₂ и x₃, полученных моделированием шагов алгоритма, совпали с результатами вычислений М.С. Бухтяка из Томского государственного университета посредством его авторской программы, использующей шаг δ (из изложенного в [1] способа) в среде математического пакета Maple 8. Поэтому справедливо следующее заключение: испытание предлагаемого алгоритма на возможность его применения к высокоточному вычислению прошло успешно.

Алгоритм, как показал компьютерный эксперимент, позволяет научно-техническим работникам без использования метода касательных оперативно и самостоятельно создавать и применять небольшие по объёму файлы для решения кубических уравнений (в частности, посредством программы EXCEL). Алгоритм не выполняет присущие способу из [1] действия α, β, γ, δ, и поэтому используют дихотомию. Эти обстоятельства приводят к следующим выводам: задача исследования решена; цель исследования достигнута.

Автор благодарит Е.М. Малекo за проявленный им интерес к модификации алгоритма на стадии подготовки данной статьи к опубликованию и за помощь, связанную с Maple. Также за помощь, связанную с Maple, автор благодарит М.С. Бухтяка. Автор благодарит В.М. Галкина из Томского политехнического университета за его помощь (при подготовке данной статьи к печати) в улучшении формулировки шага 2 алгоритма.

ЛИТЕРАТУРА

1. *Справочник по специальным функциям с формулами, графиками и таблицами*. М.: Наука, 1979. 832 с.
2. *Математический энциклопедический словарь*. М.: Сов. энциклопедия, 1988. 847 с.
3. *Сборник задач по методам вычислений / под ред. П.И. Монастырского*. – Мн: Изд-во БГУ им. В.И. Ленина, 1983. 288 с.
4. *Несмеев Ю.А.* Об одном подходе к решению алгебраических уравнений 3-й и 4-й степеней // *Вестник Томского государственного университета. Математика и механика*. 2011. № 1(13). С. 26–30.
5. *Глонти Э.Н.* Таблицы корней и квадратурных коэффициентов полиномов Якоби. М.: Вычислительный центр АН СССР, 1971. 238 с.
6. *Руководство по программированию под управлением MS DOS*. М.: Радио и связь, 1995. 544 с.

Статья поступила 15.10.2013 г.

Nesmeev Yu.A. AN ALGORITHM FOR SOLVING CUBIC EQUATIONS

An algorithm for finding the roots of a cubic equation is proposed. It uses the dichotomy and the method of undetermined coefficients. It takes into account the relative position of the parabola and the hyperbola in the plane and the location of roots of algebraic equations in the complex plane. A modification of the algorithm is proposed. Using the algorithm, three examples are solved. One of them is looking for roots of the Jacobi polynomial. The well-known numerical method for solving cubic equations is estimated (from the viewpoint of the scientific and technical personnel). Its following stages are estimated. α) Searching for an interval that contains one root. β) Calculation of the initial approximation of the root. γ) Calculation of next approximations. δ) Dividing the left side of the cubic equation by a binomial. ϵ) Calculation of roots of a quadratic polynomial. The author reported the aim of researches presented in this article. The aim was to build such a numerical method of solving the equation $x^3+k_1x^2+k_2x+k_3=0$, (1) that is available to technologists and does not use the method of tangents. The author reported the problem of the research: to construct such an algorithm that computes roots of equation (1) without performing actions α , β , γ , and δ and uses the dichotomy. The problem of the research was solved under the assumption that $k_3 \neq 0$. The used methods are as follows: studying and analyzing the literature, mathematical calculations, mathematical modeling, and computer experiment. The following algorithm is presented. **1.** Calculating values of the quantities b_3 , c_0 , n_b , and n_c by formulas $b_3 = \max \{1, |-2k_1|, |k_1^2 + k_2|\}$, $c_0 = \max \{|-2k_1|, |k_1^2 + k_2|, |-k_1k_2 + k_3|\}$, $n_b = |-k_1k_2 + k_3| / (b_3 + |-k_1k_2 + k_3|)$, and $n_c = 1 + c_0$. **2.** Checking quantities n_b for being a root of the equation $q(z) \equiv z^3 - 2k_1z^2 + (k_1^2 + k_2)z - k_1k_2 + k_3 = 0$. In the case of a positive test result, assigning the value n_b to the quantity t^* and going to step 4. **3.** Determination of the sign of the product $q(n_b) \cdot q(n_c)$. In the case of a negative sign, the dichotomy is applied to the interval $[n_b, n_c]$ and the quantity t^* is assigned the value of the found root. In the case of a positive sign, the dichotomy is applied to the interval $[-n_c, -n_b]$ and the quantity t^* is assigned the value of the found root. **4.** Calculating the real root x_1 of equation (1) by the formula $x_1 = t^* - k_1$. **5.** Evaluation of the quantities a^* and b^* by the formulas $a^* = t^*$ and $b^* = a^*(a^* - k_1) + k_2$. **6.** Calculation of roots of the equation $x^2 + a^*x + b^* = 0$ and assigning their values to the roots x_2 and x_3 of equation (1). In two examples, the author used a specially designed computer programs written in Turbo Pascal. Arithmetic operations on numbers and square rooting generation for non-negative numbers were performed by arithmetic

expressions and *sqrt* function, respectively, from the programming language standard. In one example, operations on numbers were replaced by operations on strings.

Keywords: dichotomy, complex plane, root of an equation.

NESMEEV Yurii Alekseevich (Magnitogorsk State Technical University, Magnitogorsk, Russian Federation)

E-mail: nes_ya@list.ru

REFERENCES

1. *Spravochnik po spetsial'nym funktsiyam s formulami, grafikami i tablitsami*. Moscow, Nauka Publ., 1979, 832 p. (in Russian)
2. *Matematicheskiy entsiklopedicheskiy slovar'*. Moscow, Sov. entsiklopediya Publ., 1988, 847 p. (in Russian)
3. Monastyrskiy P.I., ed. *Sbornik zadach po metodam vychisleniy*. Minsk, Izd-vo BGU im. V.I. Lenina, 1983, 288 p. (in Russian)
4. Nesmeev Yu.A. Ob odnom podkhode k resheniyu algebraicheskikh uravneniy 3-y i 4-y stepeney. *Vestnik Tomskogo gosudarstvennogo universiteta. Matematika i mekhanika*, 2011, no. 1(13), pp. 26–30. (in Russian)
5. Glonti E.N. *Tablitsy korney i kvadraturnykh koeffitsientov polinomov Yakobi*. Moscow, Vychislitel'nyy tsentr AN SSSR Publ., 1971, 238 p. (in Russian)
6. *Rukovodstvo po programmirovaniyu pod upravleniem MS DOS*. Moscow, Radio i svyaz' Publ., 1995, 544 p. (in Russian)