ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
1971 г. вып. 44

УДК 621.391:518.5.

ПРОЦЕДУРА РЕШЕНИЯ ПОЛИНОМИАЛЬНЫХ УРАВНЕНИЙ МЕТОДОМ МЮЛЛЕРА

В.С. Лозовский

Ряд прикладных задач вычислительной математики приводит к необходимости решения уравнений вида:

$$b_0 z^m + b_1 z^{m-1} + \ldots + b_m = 0, \tag{1}$$

корни которого, а также коэффициенты b_k в общем случае комплексные. Приводимая ниже процедура нахождения корней уравнения (1) использовалась для нахождения полюсов и нулей передаточной функции речевого тракта с целью получения формантного описания речевого сигнала [3]. При этом передаточная функция имеет вид:

$$H(z) = \frac{a_0 + a_1 z^{-1} + \ldots + a_r z^{-r}}{1 + b_1 z^{-1} + \ldots + b_m z^{-m}}. \tag{2}$$

Если корни знаменателя $H(z)$ имеют вид $z_k = r_k e^{j\varphi_k}$, то частота k - форманты

$$f_k = \varphi_k / 2\pi T \quad \text{гц},$$

а её ширина

$$(2\Delta f)_{0.707} = -\ln r_k / \pi T \quad \text{гц},$$

где T (сек) - период квантования.

Рассмотрение ряда алгоритмов решения уравнений (1) с точки зрения затрат машинного времени и точности заставляет обратить особое внимание на методы Рутисхаузера [4] и Мюллера [1,2].

В настоящей работе использован метод Мюллера, имеющий весьма простую схему и высокую надежность. Ниже дается вывод основных формул и построение вычислительного алгоритма, на что пришлось обратить особое внимание. *)

Итак, требуется найти m корней функции

$$f(z) = b_0 z^m + b_1 z^{m-1} + \ldots + b_m \tag{3}$$

в общем случае комплексных. Предположим, что в процессе итераций найдены три последовательных приближения очередного корня: z_3, z_2 и z_1, соответствующие значениям функции f_3, f_2 и f_1. Осуществив приближение исследуемой функции в районе корня каким-либо многочленом, можно, решив соответствующее уравнение, получить значения его комплексных и(или) вещественных корней, которые и могут быть приняты за новые значения корней (3) на очередном шаге. В качестве приближающей функции используется интерполяционный полином Лагранжа:

$$L_1(f(z)) = \frac{(z-z_2)(z-z_3)}{(z_1-z_2)(z_1-z_3)} f_1 - \frac{(z-z_1)(z-z_3)}{(z_1-z_2)(z_2-z_3)} f_2 + \frac{(z-z_1)(z-z_2)}{(z_1-z_3)(z_2-z_3)} f_3 \tag{4}$$

Перепишем (4) в виде полинома, вводя новую переменную λ. Для этого обозначим :

$$h_1 \triangleq z_1 - z_2; \quad h_2 \triangleq z_2 - z_3; \quad \lambda \triangleq \frac{z-z_1}{z_1-z_2};$$

$$\lambda_1 \triangleq h_1/h_2 = \frac{z_1-z_2}{z_2-z_3}; \quad \delta_1 \triangleq 1 + \lambda_1 = \frac{z_1-z_3}{z_2-z_3}. \tag{5}$$

Используя (5), выписываем необходимые соотношения:

$$\frac{z-z_1}{z_2-z_3} = \lambda\lambda_1; \quad \frac{z-z_3}{z_2-z_3} = \lambda\lambda_1 + \delta_1; \tag{6}$$

$$\frac{z-z_2}{z_2-z_3} = \lambda\lambda_1 + \lambda_1; \quad \frac{z-z_2}{z_1-z_2} = \lambda + 1.$$

*) Первоначально нами использовалась процедура КАМЮ решения полиномиальных уравнений, входящая в математическое обеспечение АЛЬФА-транслятора [5] и также реализующая метод Мюллера. При этом в ряде ситуаций возникали аварийные остановы. Вследствие этого, а также с целью получения более высокой точности при меньшем числе итераций схема вычислений была полностью пересмотрена.

Подставляя (6) в (4) и выполняя элементарные преобразования, находим :

$$L_1(f(z)) = \lambda^2 \frac{\lambda_1}{\delta_1} t_2 + \lambda g_1 + f_1, \tag{7}$$

где

$$t_2 = f_1 - \delta_1 f_2 + \lambda_1 f_3,$$

$$g_1 = (1 + \lambda_1/\delta_1) f_1 - \delta_1 f_2 + \lambda_1^2 f_3/\delta_1.$$

Решив уравнение (7) и получив значение $\lambda \triangleq \lambda_0$ на данном шаге, переходим к переменной z :

$$z_0 = z_1 + \lambda_0 h_1 \triangleq z_1 + \delta z. \tag{8}$$

Имеем:

$$\lambda_0 = -\frac{g_1\delta_1}{2\lambda_1 t_2} \pm \sqrt{\left(\frac{g_1\delta_1}{2\lambda_1 t_2}\right)^2 - \frac{f_1\delta_1}{\lambda_1 t_2}}. \tag{9}$$

Величина поправки :

$$\delta z = -\frac{g_1\delta_1 h_1}{2\lambda_1 t_2} \pm \sqrt{\left(\frac{g_1\delta_1 h_1}{2\lambda_1 t_2}\right)^2 - \frac{f_1 h_1^2 \delta_1}{\lambda_1 t_2}}. \tag{10}$$

Упрощая (10), получаем :

$$\frac{h_1\delta_1}{\lambda_1} = z_1 - z_3; \quad g_2 \triangleq \frac{z_1-z_3}{t_2}; \quad r \triangleq -\frac{g_1 g_2}{2};$$

$$\delta z = r \pm \sqrt{r^2 - f_1 h_1 g_2}. \tag{11}$$

При приближении к корню функции (3) естественно полагать, что из двух значений поправок δz (11) следует выбирать наименьшее по модулю, вычисляя очередное приближение корня по (8). Затем определяется значение функции (3) — f_0 в полученной точке. Осуществляя замену индексов, приходим снова к исходной ситуации, когда мы считали известными z_3, z_2, z_1 и f_3, f_2, f_1.

На начальной стадии итерационного процесса может оказаться, что вместо приближения происходит удаление от корня. Это может быть обнаружено, если вновь полученное значение f_1 станет чрезмерно большим или же в несколько раз превзойдет предыдущее значение f_2. В этом случае величина поправки для рассматривашего шага уменьшается и берется с противоположным знаком. Процесс коррекции поправки может быть повторен нужное число раз.

Критерием окончания итерационного процесса определения очередного корня может служить уменьшение значения функции ниже заданного небольшого порога или же достижение малой относительной ошибки по значению аргумента.

Полученный таким образом корень может оказаться вещественным или комплексным. Поскольку коэффициенты (3) в нашей задаче вещественны, во втором случае для нахождения очередного корня достаточно взять комплексно-сопряженную величину.

Найденные корни из анализируемого полинома (3) исключаются делением на $x - x_1$ в случае вещественных или на полином вида $x^2 + r_1 x + r_2$ в случае комплексных корней. В последнем случае $r_1 = -2a$; $r_2 = a^2 + b^2$, если один из корней имеет вид $a + jb$.

Для деления можно воспользоваться рекуррентной формулой (II) [3], переписав её в виде:

$$a'_i = a_i - \sum_{j=1}^{n} r_j a'_{i-j} \psi(j \leq i), \qquad (12)$$

где n – степень полинома – делителя,

a'_i – коэффициенты полинома – частного,

$$0 \leq i \leq m-n$$

$$\psi(x) \begin{cases} = 1, & \text{если } x = \underline{\text{истина}} \\ = 0, & \text{если } x = \underline{\text{ложь}} \end{cases}$$

Остается решить вопрос о начальных значениях аргумента: x_1, x_2 и x_3. Опыт говорит, что в общем это не очень существенно, однако процесс быстрее сходится, если приняты комплексные величины, находящиеся в области расположения корней. В нашем случае это круг единичного радиуса.

Экспериментальная проверка

С помощью описанной процедуры было решено несколько сотен уравнений. Специальной программой задавались случайные значения коэффициентов полинома. Исследовались полиномы до 14-й степени. Для получения очередного значения корня требовалось от двух до четырнадцати итераций. Полученные значения подставлялись в исходный полином. Отклонение вычисленного значения от нуля имело порядок 10^{-6} (при задании коэффициентов исходного полинома в пределах от -50 до +50). Приведенные цифры следует считать ориентировочными, поскольку и точность решения, и число

итераций зависят как от значений исходных коэффициентов, так и от близости корней.

На рис. I приведен график поиска процедурой корней полинома:

$$x^{11} + 23.277\,x^{10} - 18.364\,x^9 - 45.087\,x^8 - 13.451\,x^7 - 8.538\,x^6 + 28.010\,x^5 - 30.528\,x^4 + 47.482\,x^3 - 33.046\,x^2 - 35.564\,x - 18.61 = 0.$$

Исходная точка всех итераций: $0.7 + j0.7$. Сопряженные корни не приведены. Кроме указанных, есть ещё один вещественный корень: -23.966, не поместившийся на поле рисунка.

Описание процедуры

Процедура *Muller* написана на языке АЛЬФА и использовалась для счета на ЭВМ М-220 и БЭСМ-6 (см. Приложение).

Обращение имеет вид: *Muller* (Z, n, R); здесь $Z[0:n]$ – вещественный массив коэффициентов исходного полинома, расположенных в порядке убывания степеней, $n \leq 20$; $R[1:n]$ – комплексный массив корней. Значение Z после выполнения процедуры сохраняется.

Комплексная процедура $f(x)$ служит для вычисления текущих значений полинома.

Если абсолютная величина первых коэффициентов полинома не превышает 10^{-14}, они игнорируются и рассматривается оставшаяся часть как полином соответственно меньшего порядка.

Исходные значения аргумента: $0.7 + j0.7$, 1 и j.

Если абсолютное значение тангенса разности между аргументом корня и вещественной или мнимой осью не превышает 10^{-6}, то корень считается соответственно чисто вещественным или чисто мнимым.

Для найденного комплексного корня сразу же записывается его сопряженная пара.

Итерации по уточнению корня прекращаются при достижении функцией в найденной точке значения 10^{-9}, либо при уменьшении до величины 10^{-9} относительной разности двух последовательно найденных значений корня.

Процедура *Muller* может быть легко модифицирована для нахождения корней полинома с комплексными коэффициентами. При этом массив Z и массив a в процедуре должны быть комплекс-

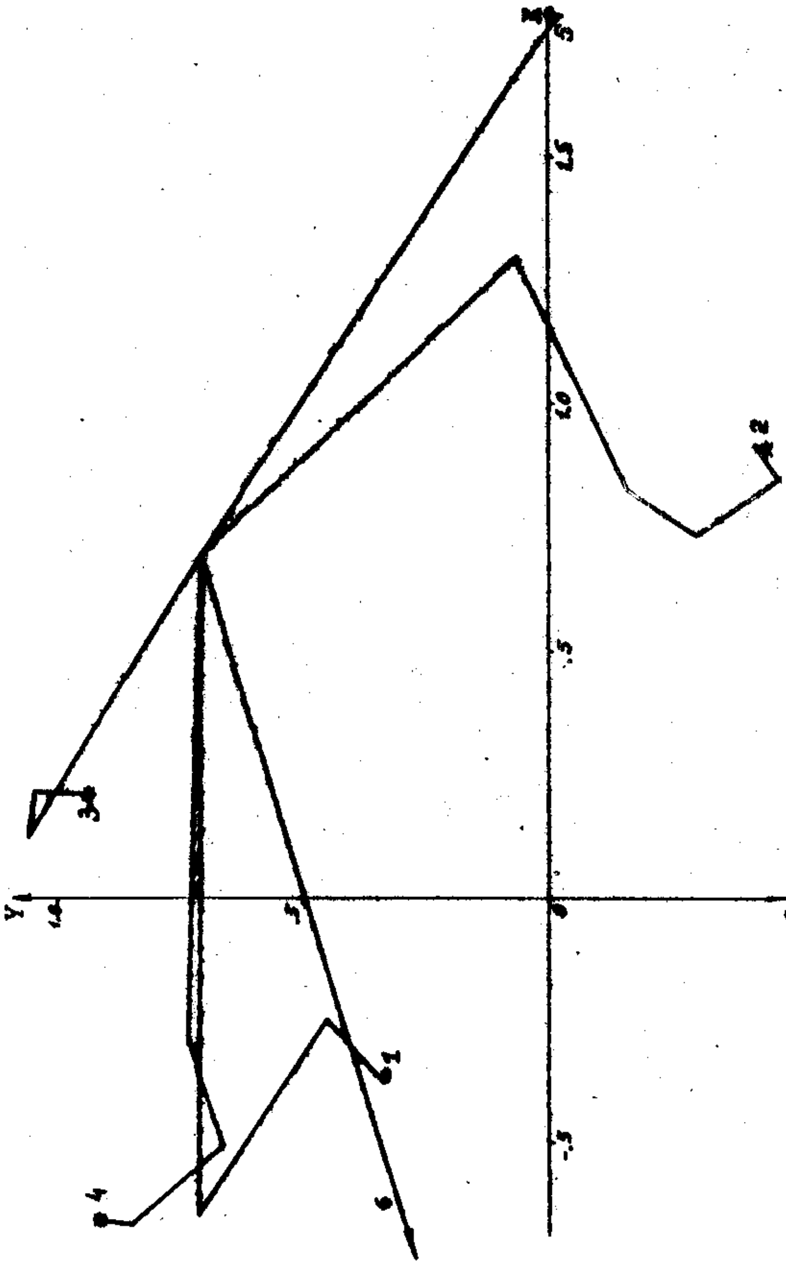

Рис.I График поиска корней полинома процедурой Muller.

ными ; из процедуры нужно убрать вычисление сопряженных корней и принудительное обнуление мнимых и вещественных компонент корня.

Автор благодарит Г.И.Кожухина за консультации по процедуре КАМЮ.

Литература

I. Muller D. E. A Method for Solving Algebraic Equations Using an Automatic Computer. Mathematical Tables and Other Aids to Computation, 10, 208, 1956.

2. Ланс Дж. Численные методы для быстродействующих вычислительных машин. М., ИЛ., 1962.

3. Лозовский В.С. Аппроксимация отклика системы в Z -плоскости и формантный анализ речи, в сб. "Вычислительные системы", Новосибирск, № 37, I969.

4. Рутисхаузер Г., Алгоритм частных и разностей. М., ИЛ.,I960.

5. Кожухин Г.И. Стандартная процедура КАМЮ. Процедура АВОСТ, сер. Стандартные программы и процедуры, вып. IO. ВЦ СО АН СССР, Новосибирск,(в печати).

Поступила в редакцию
I3.I.I97I г.

ПРИЛОЖЕНИЕ

```
001)  ПРОЦЕДУРА ¬MULLER(¬Z,N,¬R); ЗНАЧЕНИЕ N; ЦЕЛЫЙ N; НАЧАЛО ЦЕЛЫЙ Q,I,M;
002)  МАССИВ A[-1:20],B[1:2],*ТАУ*; КОМПЛ Z1,Z2,Z3,F1,F2,F3,G1,G2,*ДЕЛЬТА*,*ЛАМБДА*,H,
003)  *СИГМА*,T1,T2,T,*ЭТА*,R; *СИГМА*:=*СИГМА*1+I*СИГМА*2; *СИГМА*1:=0; *СИГМА*2:=0; T=TR+ITI;
004)  T2=C+ID; Z1=X+IY; *ЭТА*:=*ЭТА*1+I*ЭТА*2; *ЭТА*1:=0; *ЭТА*2:=1;
005)  КОМПЛ ПРОЦЕДУРА F(X); НАЧАЛО КОМПЛ Z; Z:=*СИГМА*+A[0]; ДЛЯ I:=1,...,N ЦИКЛ Z:=Z*X+A[I]
      ;
006)  F:=Z КОНЕЦ ; ДЛЯ I:=0,...,N ЦИКЛ A[I]:=¬Z[I]; Q:=0; A[-1]:=0;
007)  ¬L: ЕСЛИ N=0 ТО НА OUT; Q:=Q+1;
008)  ЕСЛИ ABS(A[0])<₁₀-14 ТО НАЧАЛО ¬R[Q]:=*СИГМА*; N:=N-1; ДЛЯ I:=0,...,N ЦИКЛ A[I]:=A[I+1];
      НА ¬L КОНЕЦ ;
009)  ЕСЛИ N=1 ТО НАЧАЛО ¬R[Q]:=*СИГМА*-A[1]/A[0]; НА OUT КОНЕЦ ; Z1:=.7*ЭТА*,7; Z2:=СИГМА*
      +1; Z3:=*ЭТА*;
010)  F1:=F(Z1); F2:=F(Z2); F3:=F(Z3); H:=Z1-Z2; *ЛАМБДА*:=H/(1-*ЭТА*);
011)  ITER: *ДЕЛЬТА*:=*ЛАМБДА*+1; G1:=(1+*ЛАМБДА*/*ДЕЛЬТА*)*F1-*ДЕЛЬТА*F2+*ЛАМБДА*×*ЛАМБДА*×F3/*ДЕЛЬТА*
      ;
012)  T2:=F1-*ДЕЛЬТА*×F2+*ЛАМБДА*×F3; ЕСЛИ C×C+D×D<₁₀-17 ТО НА ¬M; G2:=(Z1-Z3)/T2; R:=G1×G2/2;
013)  T:=R×R-F1×H×G2; *ТАУ*:=MOD(T); C:= ЕСЛИ TR+*ТАУ*<0 ТО 0 ИНАЧЕ SQRT((TR+*ТАУ*)/2);
014)  *ТАУ*:=(*ТАУ*-TR)/2; D:= ЕСЛИ *ТАУ*<0 ТО 0 ИНАЧЕ SQRT(*ТАУ*); ЕСЛИ TI<0 ТО D:=-D;
015)  T:=R+T2; T1:=R-T2; ЕСЛИ MOD(T)<MOD(T1) ТО T1:=T; Z3:=Z2; Z2:=Z1;
016)  F3:=F2; F2:=F1; ¬U: Z1:=Z2+T1; F1:=F(Z1); *ТАУ*:=MOD(F2); ЕСЛИ *ТАУ*>.01 ТО
017)  НАЧАЛО ЕСЛИ ¬RE(F1)≠0 ∨ ¬IM(F1)≠0 ТО НА ¬Q; ЕСЛИ MOD(F1)>5×*ТАУ* ТО
018)  НАЧАЛО ¬Q: T1:=-.5×T1; НА ¬U КОНЕЦ КОНЕЦ ; *ЛАМБДА*:=T1/H; H:=T1; ЕСЛИ MOD(F1)>₁₀-6 AND MOD
      (1-Z2/Z1)
019)  >₁₀-9 ТО НА ITER; ¬M: ЕСЛИ ABS(X)<ABS(Y)×₁₀-6 ТО X:=0;
020)  ЕСЛИ ABS(Y)<ABS(X)×₁₀-6 ТО Y:=0; ¬R[Q]:=Z1; ЕСЛИ Y≠0 ТО НАЧАЛО Q:=Q+1; M:=2;
021)  N:=N-1; ¬R[Q]:=¬CON(Z1); B[1]:=-2×X; B[2]:=X×X+Y×Y КОНЕЦ ИНАЧЕ НАЧАЛО M:=1; B[1]:=-X КОНЕЦ
      ;
022)  N:=N-1; ДЛЯ I:=1,...,N ЦИКЛ НАЧАЛО A[I]:=A[I]-B[1]×A[I-1]; ЕСЛИ M=2 ТО
023)  A[I]:=A[I]-B[2]×A[I-2] КОНЕЦ ; НА ¬L; OUT: КОНЕЦ ;
```