关于求連分数不完全商的算法

侯广坤　王伊成
（计算机科学系）

摘　要

本文根据连分数不完全商的分布和其它性质，提出一种较好地寻找连分数不完全商的算法，解决了连分数在算法过程中碰到的大数的计算，保证得出不完全商的精度，并已在计算机实现。

本文给出了原则上可以求出代数无理数连分数任意数目的不完全商的算法（称为“长技术”算法，以下简称《长》算法），克服计算机空间字长的限制，在微型机实现了算法，算出$\sqrt[7]{11}$的五百个不完全商和$\sqrt[5]{3}$的二百个不完全商。

一般地用$\alpha=[a_0;a_1,a_2,\cdots]$来表示连分数，$a_1,a_2\cdots$为连分数的不完全商。

设$P_n(T)=b_n^{(n)}T^d+b_1^{(n)}T^{d-1}+\cdots+b_d^{(n)}$，其中$n\geqslant 0$，次数$d\geqslant$，首项系数$b_0^{(n)}>0$，$P_n(T)\in Z[T]$。（用$Z[T]$来表示整系数多项式环，$R[T]$表示实系数多项式环）。又设$a_n>1$是属于区间$(1,\infty)$的$P_n(T)$的唯一质正无理根（可表示成$\nu\sqrt{\ }$，其中$\mu$，$\nu$为质数），$P_n(a_n)=0$，$a_n=[\alpha_n]$是使$P_n(a_n)<0$的最大整数。令$Q_n(T)=P_n(T+a_n)$，而$P_{n+1}(T)=(-T^d)\times Q_n(T^{-1})$，那末$Q_n(T)$有一个属于$(0,1)$的根。又因$P_{n+1}(T)$的根是$Q_n(T)$的根的倒数，所以$P_{n+1}(T)$有唯一的正根$a_{n+1}\in(1,\infty)$，且$a_{n+1}$是质无理根。又因为$P_{n+1}(T)$的首项系数同$Q_n(T)$的末项系数相等，即符号同$P_n(a_n)$相反，所以$P_{n+1}(T)$的首次系数$b_0^{(n+1)}>0$。这样$P_{n+1}(T)$满足前面对于$P_n(T)$而给出的所有假定条件。

因此，只要我们从$P_0(T)$开始，根据上面的条件，就可以构造多项式的无穷序列

$$P_0(T),P_1(T),\cdots,P_n(T),\cdots。$$

设这些多项式的根分别为$\alpha=\alpha_0,\alpha_1\cdots,\alpha_n,\cdots$。令$a_n=[\alpha_n]$，不难验证：$\alpha_{n+1}=(\alpha_n-a_n)^{-1}$。也就是说多项式$P_n(T)$的根的整数部分恰好等于$\alpha=\alpha_0$的连分数$\alpha=[a_0;a_1,\cdots]$的第$n$个不完全商$a_n$，$n=0,1,\cdots$。此外，在[15]得知：多项式$P_n(T)$的判别式$D(P_n(T))$对于$n\geqslant 0$时不依赖于$n$。

这样，求代数无理数的连分数不完全商的问题可归结为给定$P_n(T)$和a_n来产生$P_{n+1}(T)$，以及计算$P_{n+1}(T)$的根的整数部分a_{n+1}的问题。这是我们算法设计的主要依据。

本文1983年7月收到

我们着重考虑如下三个问题：（Ⅰ）$P_{n+1}(T)$的产生；（Ⅱ）满足条件的a_{n+1}的搜索；（Ⅲ）减少计算量的措施。

（Ⅰ）**根据$P_n(T)$和a_n来产生$P_{n+1}(T)$。**设

$$P_n(T)=\sum_{i=1}^{d} b_{i-i}^{(n)}T^i$$

$$P_{n+1}(T)=(-T^d)P(T^{-1}+a_n)=-\sum_{i=1}^{d}\left(b_{d-i}^{(n)}T^{d-}\left(\sum_{k=0}^{i}a_n^k\binom{i}{k}l^k\right)\right).$$

令$l=d-i+k$，上式可以改写为

$$\sum_{l=0}^{d}\left\{-\sum_{k=0}^{l}b_{l-k}^{(n)}a_n^k\binom{d+k-l}{k}\right\}T^l,$$

也就是说

$$b_{d-l}^{(n+1)}=-\sum_{k=0}^{l}b_{l-k}^{(n)}a_n^k\binom{d+k-l}{k},\quad l=0,1,\cdots,d.$$

令 $\binom{d+k-l}{k}=F_k$，　$k=1.2,\cdots,l$；　$F_0=1$。那末

$$b_{d-l}^{(n+1)}=-\left(b_l^{(n)}+\sum_{k=1}^{l}b_{l-k}^{(n)}a_n^k\,F_{k-1}(d+k-l)/k\right),$$

$$l=0,1,\cdots,d. \qquad (1)$$

在公式(1)中的所有数都是整数，因此$P_{n+1}(T)$的系数$b_{d-l}^{(n+1)}$也是整数，而且在计算$b_{d-l}^{(n+1)}$的过程中只运用算术四则运算。用计算机来产生$P_{n+1}(T)$的系数$b_{d-l}^{(n+1)}$是容易实现的。但是$b_{d-l}^{(n+1)}$在n足够大时是一个非常大的整数。

命题1[15] 设α为区间$(1,\infty)$中的$P_0(T)$的唯一的无理根，令$\alpha\leqslant l_n(T)^l$，$l>0$。此时多项式$P_n(T)$的系数$b_i^{(n)}\approx r^{(d-2)n}$，其中$d$为$P_0(T)$的最高次数，$r$是勒维常数，为3.27…[9]。

$P_n(T)$的系数依赖于d和n成指数型增大。例如，当计算$\sqrt[7]{11}$的1000个不完全商时，$P_{1000}(T)$的系数可达$(3.27)^{5000}$。众所周知，目前的计算机都不能用一般的方法存贮这样大的数。计算机中表示的浮点数只是在一定字长限制内的一种近似表示。从公式(1)看出，随着n的增大，计算$b_{d-l}^{(n)}$的误差变得越来越大。在我们的算例中，在十进制有效位为17位的计算机中计算$\sqrt[7]{11}$的不完全商，计算到a_{11}时，所得到的结果已不是不完全商，对于$\sqrt[6]{3}$，a_{16}已不是不完全商。因此，将一个很大的数在计算过程中完全严格地保持计算的精度是计算连分数不完全商的算法的关键。为此目的我们设计《长》算法来解决此问题。所谓《长》算法是用数组来表示一个数，这样形式的数的算术运算由特殊定义的数组间的运算来完成。考虑到我们所解决的问题和计算过程中只用到整数的加，减，乘运算的特点，我们只设计《长》加《长》乘和《长》减。

用一个数组 X表示一个所需要的多精度数。表示方法如下：设计算机的有效位为$2t$，而我们的需要的多精度数最大有效位为Mt，那末数组X就由M个元素组成；将这个多精度数的最低t位放在数组X具有最低下标的数组元素之中，次低的t位放在具有次低下标的数组元素之中，依次放完为止，并且数组X的每一个元素的符号同其表示的多精度数的符号相同。

我们发现，在将用《长》算法定义的数组算术运算应用到产生$P_{n+1}(T)$的计算过程中时，在n不是很大的情况下，数组X只在几个较低下标的元素表示了多精度数，但是运算是按$M\cdot t$位数来运算，因而增加了计算时间。为此，要求《长》算法中必须包括删除不必要的计算的功能。为此目的我们另设变量XX来指出这个多精度数占用数组X的元素的数目，算法使得数组X中元素下标大于XX的那些数组元素不参加运算，从而节省计算时间。例如：设多精度数a有$10t+k$位，$k<t, 10<M$，将a放在数组A之中。这样$A(i)$存放a的$(i-1)t+1$到it位，$i=1,\cdots,\cdots 10$；最后k位放在$A(11)$中，$AA=11$。算法过程中数组A下标大于11的数组元素不参加运算。

此外，数组间运算过程实际上是各元素单独进行的。为了解决进位问题，我们只用了计算机有效位$2t$中的t位，其它t位用来处理进位，使计算方便。特别是对于定义《长》乘法更有好处。

下面描述《长》算法，用SPARKS语言描述：

《长》加法：C = A⊕B

```
Procedure ADD(A,B,C,AA,BB,CC);
d←—0; ma←—max(AA,BB); CC←—ma;
Af←—sgn(A(AA)); Bf←—sgn(B(BB));
if Af = Bf then { for i←—1 to ma do
    C(i)←—A(i) + B(i) + d; d←—〔C(i) * 10^-t〕;
    C(i)←—C(i) - d * 10^t end
if d≠0 then CC←—ma + 1; C(cc)←—d; return}
i←—ma + 1
repeat i←—i - 1 until i = 0∧|A(i)| ≠ |B(i)|
if i = 0 then {C←—0; CC←—0; return}
case
: |A(i)| > |B(i)| : Ā←—A(1); for s←—1 to ma do
  if |A(s)| < |B(s)| then {C(s)←—Ā + B(s) + Af * 10^t;
    Ā←—A(s + 1) - Af}
  else {C(s)←—Ā + B(s); Ā←—A(s + 1)} end
: |A(i)| < |B(i)| : B̄←—B(1); for s←—1 to ma do
  if |A(s)| > |B(s)| then {C(s)←—A(s) + B̄ + Bf * 10^t;
    B̄←—B(s + 1) - Bf}
else {C(s)←—A(s) - B̄; B̄←—B(s + 1)} end
```

```
While C(i)=0 do i←—i-1 end cc←—i
end ADD
```

《长》减法：C=A⊖B。

```
procedure SUB(A,B,C,AA,BB,CC)
for i←—1 to BB do B(i)←—-B(i) end
  call ADD(A,B,C,AA,BB,CC)
for i←—1 to BB do B(i)←—-B(i) end
  end SUB
```

《长》乘法：C=A⊗B。

```
procedure MPY(A,B,C,AA,BB,CC)
  c←—0
for s←—1 to BB do
{d←—0; EE←—AA;
    for i←—1 to AA do {E(i)←—A(i)*B(s)+d;  d←—〔EE(i)*10^-t〕;
      E(i)←—E(i)-d*10^t} end
if d≠0 then (E(AA+1)←—d; EE←—AA+1)
    d←—0;
for i←—s to S+EE do {C(i)←—C(i)+E(i)+d;
    d←—〔C(i)*10^-t〕; C(i)←—C(i)-d*10^t} end end}
i←—AA+BB; While C(i)=0 do i←—i-1 end
cc—→i
end MPY
```

利用《长》算法，根据公式(1)不难产生$P_{n+2}(T)$。

(Ⅱ)找寻a_{n+1}。如前面所说，a_{n+1}是$P_{n+1}(T)$的根的整数部分，$P_{n+1}(a_{n+1})<0$，$P_{n+1}(a_{n+1}+1)>0$。因此寻找a_{n+1}的问题是在自然数集合N中搜索满足$P_{n+1}(a_{n+1})<0$，$P_{n+1}(a_{n+1}+1)>0$的数a_{n+1}。为此我们讨论a_n在N中的分布。

设$M_n(k)=\{\alpha\in(0,1):\alpha=〔a_0;a_1,a_2,\cdots〕, a_n=k\}$和$\mu$表示在线段〔0,1〕中的标准测度，此时

$$\lim_{n\to\infty}\mu(M_n(k))=\log_2\frac{(k+1)^2}{k(k+2)}.$$

根据高斯——库兹明规则，令

$$\varphi_n(k)=\{i\in N:1\leqslant i\leqslant n,\ a_i=k\},$$

$$\phi_n(k)=\{i\in N:\ 1\leqslant i\leqslant n,\ a_i\geqslant k\},\ 则$$

$$\lim_{n\to\infty}\frac{\varphi_n(k)}{n}=d(k)=\log_2\frac{(k+1)^2}{k(k+1)},$$

$$\lim_{n\to\infty}\frac{\phi_n(k)}{n}\sum_{i=0}^{\infty}d(i)=\log_2(1+\frac{1}{k})\approx\frac{1.44}{k}.$$

从上面估计看出，对于a_n进行顺序检索，比较次数的期望值较小。但是寻求a_n的过

程是将检索到的a_n代进多项式$P_n(t)$中，从而看$P_n(a_n)$和$P_n(a_n+1)$的符号变化。因此我们既要考虑检索的速度，又要考虑计算的方便。在我们的试验中表明，分区顺行检查较为有效，因10，10^2，10^3，…代入公式$P_n(T)$中较易计算，特别是对于《长》算术运算。如下是改进的期望检查的递归过程：

```
Procedure JS(A,P_{n+1}); ||参数A返回a_{n+1}的值||
  A1←——A←——1
While P_{n+1}(A)<0 do
  A1←——A; A←——A*10
end
    if P_{n+1}(A)=0 then return
A←——A1; call TG(A,A1,P_{n+1})
end JS
Procedure TG(A,AI,P_{n+1})
  for i←——1 to 10 do
    A←——AI+A
case
: P_{n+1}(A)>0; A←——A-AI; if AI=1 then return;
        AI←——1/10; Cal (TG(A,AI,P_{n+1})
: P_{n+1}(A)=0:return
end end end TG
```

这个算法实现起来并不难，但执行是非常复杂的。这主要是因为每次检索都要计算$P_n(T)$的符号。在n较大时$P_n(T)$的系数非常大。虽然在(1)中解决了产生$P_n(T)$的可能性问题，但随着n的增加，计算不完全商的时间也增加。《长》算法是以时间耗费来换取空间的精度。

加速计算的问题是一个突出的问题，下面给出一个加速计算的方法。

(Ⅱ)因为计算的时间随n的增加是由于$P_n(T)$的系数的增大。根据命题1，可以估计，在$r^{(d-2)n}$的数的有效位为$\frac{(d-2)n}{2}$。在计算$b_{d-l}^{(n)}$的过程中主要是乘法，所以使计算的次数大约为$\left(\frac{(d-2)n}{2}\right)^2/2\approx(d-2)^2n^2/8\approx n^2/8$。估计表明，计算的时间耗费以$n^2/8$速度增长。因此，加速计算的措施是减少系数计算的有效位。也就是说，寻找一个方法，使得在确定计算第n个不完全商时，只需用系数的N_n位参加计算就足可保证a_n准确，而减少不必要的计算。

为此，设$a_n^{(1)}$，$a_n^{(2)}$，…，$a_n^{(d)}$是$n\geqslant 0$时多项式$P_n(T)$的根，$a_n=a_n^{(1)}>1$，其它的根可以是复数。令$a_0^{(i)}=a^{(i)}$，$i=1,2,\cdots,d$，其中$a^{(i)}$是多项式$P_0(T)$的已知根，对于$n\geqslant 1$有$a_n^{(i)}=(a_{n-1}^{(i)}-a_n)^{-1}$，$i=1,2,\cdots,d$。

引理 [12] 存在$N>0$，对于任意$n\geqslant N$都有

$$0<-\mathrm{Re}(a_n^{(i)})\leqslant|a_n^{(i)}|<1。$$ 证明看[12]。

下面确定N，使得对于任意$n>N$都满足弱条件：$0<-\mathrm{Re}(a_n^{(i)})<2$，$i=2,3,\cdots,d$。令

$$v=\inf\{v_{i,n}=|\alpha^{(i)}-P_n/e_n|,\ 2\leqslant i\leqslant d, n>2\},$$

其中P_n/q_n为α的渐近分数，$P_n/q_n=[a_0,a_1,\cdots,a_n]$。取$n\geqslant 2$，使

$$(q_{N-1}q_{N-2}v)^{-1}<1。\tag{2}$$

从而给出 $0<-\mathrm{Re}(a_n^{(i)})<2a_{n-1}^{-1}\leqslant 2$。

如果$\alpha^{(i)}$为实根。因$\alpha^{(1)}$是$P_0(T)$的唯一正根并且$\alpha^{(1)}>1$，所以对于$n>2$有$v_{i,n}>3/4$。如果$\alpha^{(i)}$为复数根，则$v_{i,n}\geqslant|\mathrm{Im}(\alpha^{(i)}|>0$。　于是

$$v\geqslant\min\ (\min\ (|\mathrm{Im}(\alpha^{(i)}|\neq 0),\ 3/4)。\tag{3}$$

命题2 设$P_0(T)\in Z[T]$，且满足前所给予它的条件。设$N>2$由引理给出。对于$n\geqslant N$，令$F(T)=(1/b_0^{(n)})P_n(T)=T^d+b_1'T^{d-1}+\cdots+b_d'$，其中$b_0^{(n)}$为$P_n(T)$的首项系数。设$\alpha=a_n^{(1)}$为$F(T)$的质根，并设对于$a_n^{(1)}$存在$0<\delta_0<1$，使得$\min(\alpha-[\alpha],[\alpha]+1-\alpha)>\delta_0$，并且取$e=\delta_0/(d+1)H^d$，其中$H$为多项式的高。

此时，对于任意$Q\in R[T]$，$Q(T)=C_0T^d+C_1T^{d-1}+\cdots+C_d$，设$\beta>1$是$Q(T)$的最小正根，只要满足$|b_i'-C_i|<e, i=0,1,\cdots,d$，则$[\beta]=[\alpha]$。

证明 根据多项式高的定义，$F(T)$的高$H=1+\sum_{i=1}^{d}|b_i'|$。显然$H>1$，并且$F(T)$的所有实根都位于区间$(-H,H)$中。设对于引理中的确定的N，$F(T)=(1/b_0^{(n)})P_n(T)$满足$n\geqslant N$。设$r_1,r_2,\cdots,r_d$是$F(T)$的根，且$\alpha=r_1$。根据引理，对于$i=2,3,\cdots d$，$\mathrm{Re}(r_i)<0$。从而得到对于$i>1$和$x\in[1,\infty)$，$|x-r_i|>1$。于是$|F(x)|=\prod|x-r_i|>|x-r_1|=|x-\alpha|$，$x\in(1,\infty)$。现将$Z=[d]\geqslant 1$和$Z+1$分别代入上式，由$\delta_0$的定义得：

$$|F(Z)|>|Z-d|>\delta_0;\quad |F(Z+1)|>\delta_0。\tag{4}$$

对于$F(T)$求导，$F^1(T)=dF(T)/dT$。被$r_k'\in C$(C—复数域)，$k=1,\cdots,d-1$，是$F^1(T)$的根。如果$r_k'\geqslant 1$和$r_k'\notin(Z,Z+1)$，则

$$|F(r_k')|=\prod|r_k'-r_i|>|r_k-r_k'-\alpha|>\delta_0。\tag{5}$$

这表明在集合$[1,\infty)-[Z,Z+1]$中，$F(T)$的绝对值的理想极值以δ_0为下界。

对于任意的$Q(T)\in R[T]$，$Q(T)=C_0T^d+\cdots+C_d$，并满足$|C_r-b_r'|<e$，$i=0,1,\cdots,d$，有

$$|F(T)-Q(T)|=(T^d+\cdots+b'_d-(C_0T^d+\cdots+C_d)|$$

$$\leqslant|(1-C_0)+(b'_1-C_1)+\cdots+(b'd-Cd)|T^d<(d+1)\varepsilon T^d\leqslant\delta_0,\ T\in[1,H]。\qquad(6)$$

不等式(4)、(5)、(6)表明，$Q(T)$没有属于$[1,H]\setminus(Z,Z+1)$的根。另一方面，从不等式(4)、(6)可以看出$F(Z)$和$Q(Z)$同号，$F(Z+1)$与$Q(Z+1)$也同号。这是因为$|F(Z)|>\delta_0$，而$|F(Z)-Q(Z)|\leqslant\delta_0$。于是有：$Q(Z)<0$；$Q(Z+1)>0$也就是说在区间$(Z,Z+1)$至少有多项式$Q(T)$的一个根$\beta$，即$[\beta]=[\alpha]$。〔证毕〕

(二) 应用命题2可以减少a_n的计算量。

首先利用(3)，(2)选取v和N，并设$n\geqslant N$。设$P_{n+1}(T)\in Z[T]$，$P_{n+1}(T)$的首项系数$b_0^{(n+1)}\geqslant1$，$F(T)=(1/b_0^{(n+1)})P_{n+1}(T)$满足命题2条件。此时如果选取$\delta_0$充分小，能使$a_{n+2}<\delta_0^{-1}$，$\delta_0$就能满足命题条件，$\min(\alpha-[\alpha],[\alpha]+1-\alpha)>\delta_0$。(见文献[15])。从命题可以得出对于$F(T)$的小数部分只需要保留$n_{10}=[\log_{10}\varepsilon^{-1}]+1=[\log_{10}(d+1)H^d\delta_0^{-1}]+1$位，可将后面的若干小数位舍弃，这样得到的多项式$\overline{F}(T)$与$F(T)$的对应系数之差的绝对值小于$\varepsilon$，因而它们质正根整数部分相同。这样$P_{n+1}(T)$的整系数可以按下列公式取整：

$$N_{10}=m_{10}+n_{10}\qquad(7)$$

其中$m_{10}=[\log_{10}B/b_0^{(n+1)}]$，$B=\max\{|b_i^{(n+1)}|,\ 0\leqslant i\leqslant d\}$。

不难看出，m_{10}是$F(T)$的最大系数中整数部分所占的位数。这样，在计算$P_{n+1}(T)$的符号时，只要将其系数按$N_{10}-si$取整进行计算就可以了，其中$s_i=[\log_{10}B]-[\log_{10}bi]$，$i=0,1,\cdots,d$，即最大系数与第$i$个系数之差。

为了在计算H时不使用《长》技术。令$\bar{H}=(d+1)\delta_0^{-m}$，其中$m\geqslant1$且满足下列条件：对于$P_n(T)$的任意系数与其最大系数之比的绝对值小于$\delta_0^{-m}$，这里$n\geqslant N$。要检验这个条件只要检验多项式$P_n(T)$的系数的阶就可以了。显然$H\leqslant\bar{H}$，令$H=\bar{H}$不影响命题2的结果，将$H=\bar{H}$代入(7)得：

$$N_{10}=[m\log_{10}\delta_0^{-1}]+[(d+1)\log_{10}(d+1)+(dm+1)\log_{10}\delta_0^{-1}]+1。\qquad(8)$$

例：若$H\leqslant(d+1)\delta_0^{-m}$，$\delta_0=10^{-4}$，$m=2$，$d=3$，则由(8)得38。

这样要计算多项式$P_0(T)$的唯一正根的M个最初不完全商，必须进行下列步骤：

(1)根据(3)计算常数$v=v(P_0(T))$，并根据它和(2)式寻找整数$N>2$。对于$n<N$的不完全商a_n，可以通过$P_n(T)$的系数值计算得到。

(2)对于$n\geqslant N$的不完全商a_n，先选取δ_0，使其适用于所有的$P_n(T)$，其中$N\leqslant n\leqslant M$。也就是选取δ_0对于$N\leqslant n\leqslant M$，使$a_n<\delta_0^{-1}$。然后借助于多项式$\bar{P}_n(T)$来计算不完全商a_n，其中$\bar{P}_n(T)$是将$P_n(T)$的系数按$N_{10}-s_i$取整所得到的多项式，$i=0,1,\cdots,d$。

(3)检查所计算的不完全商的正确性。将a_{n+1}和$a_{n+1}+1$分别代入$P_n(T)$，并检查在此区间$P_n(T)$的符号是否有变化。若变化，则a_{n+1}计算正确，否则a_{n+1}的计算是不正确的，也就是δ_0选取不够小。这时只要令$\delta_0=\delta_0\times10^{-1}$，再进行(2)的计算。

利用这个方法并也还需要使用《长》算法，但总的来讲减少了计算量。

利用文中的算法，在微型机上算出了$\sqrt[7]{11}$的500个不完全商和$\sqrt[5]{3}$的200个不完全商。

参考文献

〔1〕J.Von. Neumann, B. Tuckerman, Continued Fraction Expansion of 2—MTAC, 9, p23-24.

〔2〕R.D.Richymya, M. Deveaney, N. Metropalis, Continued Fraction Expansions of Algebraic Mumbers—Number, Math., 1962,4, P68-84.

〔3〕А. Д. Брюно Разложение абъебраигсских чисел В цепные дроби, журн. Восгислит. Матем. физики, 1964,4, № 2 С211-221.

〔4〕R.E.Churckhouse, S.T.E.Muir, Continued Fractions Algebraic Numbers and Modular Invariants—J. Inst. Math. Appl., 1969. 5, P318-328.

〔5〕Х.М. Старк, Оъ'яснение некоторых жзотигеских неирерывных дробей, найденних броллхартом-В кн Бигисления В алгебре и теории чисел, «мир», 1967.

〔6〕S. Laug, H. Tratter, Continued Fraction Algarithm for Some Algebraic Humbers—J. Reine and Angew. Math., 1972, 255, P112-134.

〔7〕R.D.Richtmger, Adv, Math., 1975,3, P262-367.

〔8〕С. Ленг. Введение В теовию диофантовсех приближений, м«мир»,1970.

〔9〕А. я. хинчин. цепные дроби. п. сфизматгу, 1961.

〔10〕К.ф. Рот. Рациональные приближения алгебраигеских чисел, Математ.п-ка. 1957,1,С3-•8.

〔11〕С. ленч. Алчебра,и«мир»,1968.

〔12〕D.Contor, P.Galyeam, H.Zmmer, A Continued Fraction Algarithm for Veal Algebrai c Numbers—Math., Coup., 76 (1972), P788-797.

〔13〕D.H.lehmer. Euclid's Algarithm for Largl Numbers—Awev, Math., Uonthlg, 45(1938), P227-233.

〔14〕W.Adams, Asimptotis Diofaute Appraximation to e.proc. Not Acad sci D. S.A., 1966, 55, P28-31.

〔15〕А.г. Александров, Исследование на ЭВМ непрерывных дробей Б сборн. Алго ритмическис исследования вкомбинаторика изд, «наука», 1978.

〔16〕闵嗣鹤，严士健，《初等数论》，P136—148.

〔17〕Aho.A.V,J.E.Hopcroft and J.D.Ullman, The Design and Analysis of Computer Alg orithms, Addison—wesley, Reading, Mass.

An Algorithm to Find Continued Fraction Completeless Quotient

Hou Guangkun *Wang Yicheng*

Abstract

According to the distribution and other properties of continued fraction completeless quotient, this paper giver a better algorithm to find continued fraction completeless quotient. In order to solve the large number problem which appears in the algorithm and guarantee the accuracy of the completeless quotient which is found, this payer designs a 'Length Technique' algorithm. The above algorithms were successfully performed in a computer.