Computers Math. Applic. Vol. 21, No. 2-3, pp. 1–6, 1991
Printed in Great Britain.

0097-4943/91 $3.00 + 0.00

AN O(log n) ALGORITHM FOR COMPUTING PERIODIC CONTINUED FRACTIONS AND ITS APPLICATIONS

S.S. LIN and F.C. LIN
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan 10764, R.O.C.

(Received September, 1989 and in revised form January, 1990)

Abstract—An $O(\log n)$ algorithm for computing the nth convergents of periodic continued fractions is presented. It can be applied to solve the second-order linear recurrences with constant coefficients very efficiently. We also use it to approximate the quadratic numbers in $O(\log m)$ time for a result with m-digit precision. Our method can be thought as a generalization of the matrix-vector approach for computing the Fibonacci numbers. It is easy to implement because there are only some matrix multiplications and a division operation involved.

1. INTRODUCTION

Following is the nth convergent of a continued fraction:

$$a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{\cdots + \cfrac{b_{n-1}}{a_{n-1} + \cfrac{b_n}{a_n}}}}$$

which can be expressed by a more compact form:

$$a_1 + \frac{b_2}{a_2+}\frac{b_3}{a_3+}\cdots\frac{b_{n-1}}{a_{n-1}+}\frac{b_n}{a_n}$$

A straightforward computation of this continued fraction takes $O(n)$ time. In Section 2 of this paper, we will transform the computation into a sequence of matrix multiplications followed by a division operation.

A continued fraction with period p can be written as

$$a_1 + \frac{b_2}{a_2+}\frac{b_3}{a_3+}\cdots\frac{b_k}{a_k+}\frac{b_{k+1}}{a_{k+1}+}\frac{b_{k+2}}{a_{k+2}+}\cdots\frac{b_{k+p}}{a_{k+p}+}\frac{b_{k+1}}{a_{k+1}+}\frac{b_{k+2}}{a_{k+2}+}\cdots\frac{b_{k+p}}{a_{k+p}+}\frac{b_{k+1}}{a_{k+1}+}\frac{b_{k+2}}{a_{k+2}+}\cdots$$

for some constant k. Due to the periodic feature in the expression, a major part of its computation can be greatly reduced. An $O(\log n)$ algorithm for computing the nth convergent of the periodic continued fraction will be presented in Section 3. (Unless otherwise stated in the rest of the paper, all logorithms are of base 2.) In Section 4, we will illustrate some important applications of this algorithm, namely computing the Fibonacci numbers, solving the second-order linear recurrences with constant coefficients, and approximating the quadratic numbers.

This work was supported in part by National Science Council of the Republic of China under Contract NSC 77-0408-E002-09.

2. A MATRIX-VECTOR APPROACH

Consider the following series of finite continued fractions:

$$\begin{aligned}
C_1 &= a_1 + \frac{b_2}{a_2+}\frac{b_3}{a_3+}\cdots\frac{b_{n-1}}{a_{n-1}+}\frac{b_n}{a_n} \\
C_2 &= a_2 + \frac{b_3}{a_3+}\frac{b_4}{a_4+}\cdots\frac{b_{n-1}}{a_{n-1}+}\frac{b_n}{a_n} \\
&\vdots \\
C_{n-2} &= a_{n-2} + \frac{b_{n-1}}{a_{n-1}+}\frac{b_n}{a_n} \\
C_{n-1} &= a_{n-1} + \frac{b_n}{a_n} \\
C_n &= a_n
\end{aligned}$$

Let the rational form of C_i be U_i/L_i. Then, for $1 \le i \le n-1$,

$$\begin{aligned}
C_i &= a_i + b_{i+1}/C_{i+1} \\
&= a_i + b_{i+1}/(U_{i+1}/L_{i+1}) \\
&= (a_i U_{i+1} + b_{i+1}L_{i+1})/U_{i+1}
\end{aligned}$$

Therefore, $L_i = U_{i+1}$ and $U_i = a_i U_{i+1} + b_{i+1}L_{i+1} = a_i U_{i+1} + b_{i+1}U_{i+2}$.

The above second-order linear recurrence for U_i is equivalent to the following equation:

$$\begin{pmatrix} U_i \\ U_{i+1} \end{pmatrix} = \begin{pmatrix} a_i & b_{i+1} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} U_{i+1} \\ U_{i+2} \end{pmatrix}$$

In particular,

$$\begin{aligned}
\begin{pmatrix} U_1 \\ U_2 \end{pmatrix} &= \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} U_2 \\ U_3 \end{pmatrix} \\
&= \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} U_3 \\ U_4 \end{pmatrix} \\
&\vdots \\
&= \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{n-2} & b_{n-1} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} U_{n-1} \\ U_n \end{pmatrix} \\
&= \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{n-2} & b_{n-1} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{n-1}a_n + b_n \\ a_n \end{pmatrix} \\
&= \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{n-2} & b_{n-1} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{n-1} & b_n \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_n \\ 1 \end{pmatrix}
\end{aligned}$$

After we obtain U_1 and U_2, the continued fraction C_1 can be computed as

$$C_1 = U_1/L_1 = U_1/U_2$$

The computation of a finite continued fraction is thus transformed into some matrix multiplications followed by a division operation. The $n-1$ matrix multiplications involved can be performed in any order because matrix multiplication is associative.

3. COMPUTING PERIODIC CONTINUED FRACTIONS

By the computation scheme introduced in the previous section, a periodic continued fraction with period p can be approximated as follows:

$$P_n = a_1 + \frac{b_2}{a_2+}\frac{b_3}{a_3+}\cdots\frac{b_k}{a_k+}\frac{b_{k+1}}{a_{k+1}+}\cdots\frac{b_{k+p}}{a_{k+p}+}\cdots\frac{b_{k+1}}{a_{k+1}+}\cdots\frac{b_{n-1}}{a_{n-1}+}\frac{b_n}{a_n}$$

$$\begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_k & b_{k+1} \\ 1 & 0 \end{pmatrix}$$
$$\left[\begin{pmatrix} a_{k+1} & b_{k+2} \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{k+p-1} & b_{k+p} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{k+p} & b_{k+1} \\ 1 & 0 \end{pmatrix}\right]^{\lfloor\frac{n-k-1}{p}\rfloor}$$
$$\begin{pmatrix} a_{k+1} & b_{k+2} \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{k+(n-k-1)\bmod p} & b_{k+1+(n-k-1)\bmod p} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{k+1+(n-k-1)\bmod p} \\ 1 \end{pmatrix}$$

and $P_n = U_1/U_2$.

The first $k-1$ matrix multiplications

$$\begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_k & b_{k+1} \\ 1 & 0 \end{pmatrix}$$

take $O(k)$ time. The $p-1$ matrix multiplications inside the square bracket

$$\begin{pmatrix} a_{k+1} & b_{k+2} \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{k+p-1} & b_{k+p} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{k+p} & b_{k+1} \\ 1 & 0 \end{pmatrix} = M$$

take $O(p)$ time.

Now we consider the time for computing the $\lfloor\frac{n-k-1}{p}\rfloor^{th}$ power of the matrix M. Let the binary representation of $\lfloor\frac{n-k-1}{p}\rfloor$ be $d_q d_{q-1}\ldots d_1 d_0$, i.e., $\lfloor\frac{n-k-1}{p}\rfloor = \sum_{i=0}^{q} d_i 2^i$. Then $M^{\lfloor\frac{n-k-1}{p}\rfloor}$ can be expressed as $\prod_{\substack{0\le i\le q \\ d_i=1}} M^{2^i}$. For example, if $\lfloor\frac{n-k-1}{p}\rfloor = 13$ then $d_3d_2d_1d_0 = 1101$ and $M^{13} = M^1M^4M^8$. The computation of the $\lfloor\frac{n-k-1}{p}\rfloor$-th power of M can actually be completed in $0\left(\log\lfloor\frac{n-k-1}{p}\rfloor\right) = O(\log n)$ time by scanning the binary bits of $\lfloor\frac{n-k-1}{p}\rfloor$ [3].

The last $(n-k-1) \bmod p$ matrix multiplications

$$\begin{pmatrix} a_{k+1} & b_{k+2} \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{k+(n-k-1)\bmod p} & b_{k+1+(n-k-1)\bmod p} \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_{k+1+(n-k-1)\bmod p} \\ 1 \end{pmatrix}$$

take $O(p)$ time. Therefore, totally, it takes $O(k) + O(p) + O(\log n) + O(p) = O(\log n)$ time to accomplish the computation. It also can be checked that the space used is only $O(1)$.

THEOREM 1. The nth convergent of a periodic continued fraction can be computed in $O(\log n)$ time using $O(1)$ space.

4. APPLICATION EXAMPLES

In this section, we illustrate some applications of our approach to the computation of continued fractions.

EXAMPLE 1 (FIBONACCI NUMBERS). The Fibonacci numbers are defined by

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2} \text{ for } n \ge 2$$

To apply our method, let

$$\begin{aligned} C_1 &= \frac{F_n}{F_{n-1}} = 1 + \frac{1}{\frac{F_{n-1}}{F_{n-2}}} \\ C_2 &= \frac{F_{n-1}}{F_{n-2}} = 1 + \frac{1}{\frac{F_{n-2}}{F_{n-3}}} \\ &\vdots \\ C_{n-2} &= \frac{F_3}{F_2} = 1 + \frac{1}{\frac{F_2}{F_1}} \\ C_{n-1} &= \frac{F_2}{F_1} = 1 + \frac{0}{1} \\ C_n &= 1 \end{aligned}$$

$C_1 = 1 + \frac{1}{1+}\frac{1}{1+}\frac{1}{1+}\cdots\frac{1}{1+}\frac{0}{1}$ is a continued fraction with $a_1 = a_2 = \cdots = a_n = b_2 = b_3 = \cdots = b_{n-1} = 1$ and $b_n = 0$. By our method,

$$\begin{aligned} \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} &= \begin{pmatrix} a_1 & b_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} a_{n-1} & b_n \\ 1 & 0 \end{pmatrix}\begin{pmatrix} a_n \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\cdots\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned}$$

Although C_1 is not "purely" periodic because of the last term $\frac{0}{1}$, the time to compute F_n is clearly $O(\log n)$, since

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1}$$

can be computed in $O(\log n)$ time. The formula we have derived for F_n is similar to those in [1,4,6,7]. However, our approach provides another point of view which is applicable to a more general case as described in the next example.

EXAMPLE 2 (SECOND-ORDER LINEAR RECURRENCE). The second-order linear recurrence with constant coefficients is defined by

$$x_n = r \cdot x_{n-1} + s \cdot x_{n-2}$$

with given x_0 and x_1. Now for applying our method, let

$$\begin{aligned} C_1 &= \frac{x_n}{x_{n-1}} = r + \frac{s}{\frac{x_{n-1}}{x_{n-2}}} \\ C_2 &= \frac{x_{n-1}}{x_{n-2}} = r + \frac{s}{\frac{x_{n-2}}{x_{n-3}}} \\ &\vdots \\ C_{n-2} &= \frac{x_3}{x_2} = r + \frac{s}{\frac{x_2}{x_1}} \\ C_{n-1} &= \frac{x_2}{x_1} = r + \frac{s \cdot x_0}{x_1} \\ C_n &= x_1 \end{aligned}$$

$C_1 = r + \frac{s}{r+}\frac{s}{r+}\frac{s}{r+}\cdots\frac{s}{r+}\frac{s\cdot x_0}{x_1}$ is a continued fraction with $a_1 = a_2 = \cdots = a_{n-1} = r$, $a_n = x_1$, $b_2 = b_3 = \cdots = b_{n-1} = s$ and $b_n = s \cdot x_0$. The desired x_n can be computed as follows:

$$\begin{aligned}\begin{pmatrix} x_n \\ x_{n-1}\end{pmatrix} &= \begin{pmatrix} U_1 \\ U_2\end{pmatrix} = \begin{pmatrix} a_1 & b_2 \\ 1 & 0\end{pmatrix}\begin{pmatrix} a_2 & b_3 \\ 1 & 0\end{pmatrix}\cdots\begin{pmatrix} a_{n-1} & b_n \\ 1 & 0\end{pmatrix}\begin{pmatrix} a_n \\ 1\end{pmatrix} \\ &= \begin{pmatrix} r & s \\ 1 & 0\end{pmatrix}\begin{pmatrix} r & s \\ 1 & 0\end{pmatrix}\cdots\begin{pmatrix} r & s \\ 1 & 0\end{pmatrix}\begin{pmatrix} r & s\cdot x_0 \\ 1 & 0\end{pmatrix}\begin{pmatrix} x_1 \\ 1\end{pmatrix} \\ &= \begin{pmatrix} r & s \\ 1 & 0\end{pmatrix}^{n-2}\begin{pmatrix} r\cdot x_1 + s\cdot x_0 \\ x_1\end{pmatrix} \\ &= \begin{pmatrix} r & s \\ 1 & 0\end{pmatrix}^{n-1}\begin{pmatrix} x_1 \\ x_0\end{pmatrix}\end{aligned}$$

THEOREM 2. The second-order linear recurrence with constant coefficients can be solved in $O(\log n)$ time using $O(1)$ space.

EXAMPLE 3 (QUADRATIC NUMBERS). The quadratic numbers, which have the form $\sqrt{N}$ where N is a positive integer, can be approximated by the regula falsi method, the secant method, the Newton–Raphson method, and so on [2]. These methods employ an initial approximation to the root of a given equation and iteratively improve the result until the desired accuracy is achieved. However, sometimes convergence is not assured, or the convergence to the root is slow.

In applying our method, the quadratic numbers are first converted into periodic continued fractions. For example,

$$\sqrt{2} = 1 + (\sqrt{2} - 1) = 1 + \frac{1}{\sqrt{2}+1} = 1 + \frac{1}{2 + (\sqrt{2}-1)} = 1 + \frac{1}{2+}\frac{1}{2+}\frac{1}{2+}\cdots$$

So, for $\sqrt{2}$,

$$\begin{pmatrix} U_1 \\ U_2\end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0\end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0\end{pmatrix}^{n-2}\begin{pmatrix} 2 \\ 1\end{pmatrix}$$

In fact, all the quadratic numbers can be converted into simple periodic continued fractions which have the form

$$P = a_1 + \frac{1}{a_2+}\cdots\frac{1}{a_k+}\frac{1}{a_{k+1}+}\cdots\frac{1}{a_{k+p}+}\frac{1}{a_{k+1}+}\cdots\frac{1}{a_{k+p}+}\cdots\frac{1}{a_{k+1}+}\cdots\frac{1}{a_{k+p}+}\cdots$$

where $a_i \geq 1$ for all i [5]. Based on our approach, the quadratic numbers can be approximated by

$$\begin{aligned}P_n &= a_1 + \frac{1}{a_2+}\cdots\frac{1}{a_k+}\frac{1}{a_{k+1}+}\cdots\frac{1}{a_{k+p}+}\frac{1}{a_{k+1}+}\cdots\frac{1}{a_{k+p}+}\cdots\frac{1}{a_{k+1}+}\cdots\frac{1}{a_{n+1}+}\frac{1}{a_n} \\ &= U_1/U_2\end{aligned}$$

Now let us analyze the speed of the algorithm in terms of the number of precision digits computed. With our notations, the well-known rate of convergence of the continued fraction [5] can be expressed as

$$|P - P_n| < 1/U_2^2, \quad n \geq 1$$

Since

$$\begin{pmatrix} U_1 \\ U_2\end{pmatrix} = \begin{pmatrix} a_1 & 1 \\ 1 & 0\end{pmatrix}\cdots\begin{pmatrix} a_{n-1} & 1 \\ 1 & 0\end{pmatrix}\begin{pmatrix} a_n \\ 1\end{pmatrix}$$

$$\begin{aligned}U_2 &= a_2a_3\cdots a_n + \text{ lower terms} \\ &> a_2a_3\cdots a_n \\ &= a_2a_3\cdots a_k(a_{k+1}a_{k+2}\cdots a_{k+p})^{\lfloor\frac{n-k}{p}\rfloor}a_{k+1}a_{k+2}\cdots a_{k+(n-k)\bmod p} \\ &> (a_{k+1}a_{k+2}\cdots a_{k+p})^{\lfloor\frac{n-k}{p}\rfloor}\end{aligned}$$

So,

$$|P - P_n| < 1/U_2^2 < 1/(a_{k+1}a_{k+2}\cdots a_{k+p})^{2\lfloor\frac{n-k}{p}\rfloor}$$

If the required number of precision digits is m, we can let

$$n = k + p + p \cdot m/[2\log_{10}(a_{k+1}a_{k+2}\cdots a_{k+p})]$$

In this case,

$$\begin{aligned}|P - P_n| &< 1/(a_{k+1}a_{k+2}\cdots a_{k+p})^{2\lfloor\frac{n-k}{p}\rfloor} \\ &= 1/(a_{k+1}a_{k+2}\ldots a_{k+p})^{2\lfloor\frac{m}{2\log_{10}(a_{k+1}\cdots a_{k+p})}+1\rfloor} \\ &< 1/(a_{k+1}a_{k+2}\cdots a_{k+p})^{m/\log_{10}(a_{k+1}\cdots a_{k+p})} \\ &= 1/10^m\end{aligned}$$

The above inequality means the number of precision digits achieved is m. Since P_n can be computed in $O(\log n) = 0(\log(k + p + p \cdot m/(2\log_{10}(a_{k+1}a_{k+2}\cdots a_{k+p})))) = O(\log m)$ time, we have the following theorem.

THEOREM 3. The quadratic numbers can be approximated with m-digit precision in $O(\log m)$ time using $O(1)$ space.

REFERENCES

1. D. Gries and G. Levin, Computing Fibonacci numbers (and similarly defined functions) in log time, *Inform. Proc. Lett.* **11(2)**, 68–69 (1980).
2. Y. Jaluria, *Computer Methods for Engineering*, Allyn and Bacon, Inc., (1988).
3. D.E. Knuth, *The Art of Computer Programming*, 2nd edition, Addison-Wesley, Reading, MA, (1981).
4. A.J. Martin and M. Rem, A representation of the Fibonacci algorithm, *Inform. Proc. Lett.* **19(2)**, 67–68 (1984).
5. C.D. Olds, *Continued Fractions*, Random House, The L.W. Singer Company, New York, (1963).
6. F.J. Urbanek, An 0(log n) algorithm for computing the nth element of the solution of a difference equation, *Inform. Proc. Lett.* **11(2)**, 66–67 (1980).
7. T.C. Wilson and J. Shortt, An 0(log n) algorithm for computing general order-k Fibonacci numbers, *Inform. Proc. Lett.* **10(2)**, 68–75 (1980).