STUDIA UNIV. "BABEŞ–BOLYAI", MATHEMATICA, Volume **LI**, Number 2, June 2006

REMARKS ON SOME RECURRENCE RELATIONS IN LIFE ANNUITY

IOANA CHIOREAN AND CLAUDIA STAN

Abstract. The problem of life annuity is one of the main items in the insurance theory. That's why, their computation is very important. The purpose of this paper is to give a possible parallel implementation for such a computation, by using some recurrence relations and the double recursive technique.

1. Introduction

A financial operation connected to the insurance of a person has a random character, both from the insurance institution part and from the insured person.

A fundamental principle for any kind of life insurance is the one of financial equilibrium: the mean value of the gain (of the insured person) has to be equal with the mean value of the gain of the institution which made that life assurance. This value is called **insurance premium**.

2. Life payments

According with [1], [3], [4], a life payment takes place only if the insured person is alive. It can be made both by the insurance institution and by the insured. In what follows, we want to determine the insurance premium that x aged old person has to pay, to get a certain amount of money after n years (if he'll be still alive). Keeping the ideas of [3], we denote by:

$p(x, x+n)$ the probability that a x aged old person to be alive after n years

Received by the editors: 14.11.2005.

2000 *Mathematics Subject Classification.* 91B30, 65Y05, 11C20.

Key words and phrases. Insurance, parallel computation, matrices.

$q(x, x+n) = 1 - p(x, x+n)$ the probability that a x aged person to be dead after n years

l_x the survival function, it means the mean value of the number of people considered who has the chance to live at the age x.

The formula is:

$$p(x, x+n) = \frac{l_{x+n}}{l_n}. \tag{1}$$

Then, denoting by v^n the discounted present value (where v is the discounter present factor from the compound interest), we get

$$M(x) = \frac{l_{x+n}}{l_x} v^n + \left(1 - \frac{l_{x+n}}{l_x}\right) \cdot 0. \tag{2}$$

For computation, in order to use some predefined tables, some notations are also made:

$$D_x = l_x \cdot v^x. \tag{3}$$

Note. D_x are called commutation numbers.

3. **Payments in life anuities**

In order to determine the amount of money that a x aged old person has to pay, once, to get one monetary unit per year, during all his life, we denote by

a_x the mean value of the posticipated life anuity (payed at the end of every year).

According with [3], and using the commutation numbers, we get the relation:

$$a_x = \frac{D_{x+1}}{D_x} + \frac{D_{x+2}}{D_x} + \cdots + \frac{D_{x+n}}{D_x} + \ldots \tag{4}$$

If we denote by

$$N_x = D_x + D_{x+1} + \cdots + D_\omega,$$

where ω is the age when the last person of the considered generation dies, it results that:

$$a_x = \frac{D_x + \cdots + D_\omega}{D_x} = \frac{N_{x+1}}{D_x}. \tag{5}$$

4. **Recurrence relations in life anuities**

Taking into account (5), and replacing x by $x+1$, we get

$$a_{x+1} = \frac{N_{x+2}}{D_{x+1}}. \tag{6}$$

But

$$N_x = D_x + D_{x+1} + \cdots + D_\omega$$

and

$$N_{x+1} = D_{x+1} + D_{x+2} + \cdots + D_\omega.$$

Subtracting (6) from (5), we get

$$D_x = N_x - N_{x+1}$$

$$D_{x+1} = N_{x+1} - N_{x+2}$$

and replacing in (5), the relation is, successively:

$$a_x = \frac{D_{x+1} + N_{x+2}}{D_x}$$

$$= \frac{D_{x+1}}{D_x}\left(1 + \frac{N_{x+2}}{D_{x+1}}\right) = \frac{D_{x+1}}{D_x}(1 + a_{x+1}). \tag{7}$$

Because

$$\frac{D_{x+1}}{D_x} = \frac{l_{x+1}v^{x+1}}{l_x v^x} = \frac{l_{x+1}}{l_x}v = p(x, x+1)v,$$

replacing in (7), we get the recurrence relation:

$$a_x = p(x, x+1) \cdot v \cdot (1 + a_{x+1}). \tag{8}$$

Or, making some calculation, we can write

$$a_{x+1} = \frac{a_x - p(x, x+1)v}{p(x, x+1)v} = \frac{1}{p(x, x+1)}a_x - 1. \tag{9}$$

5. **Parallel computation**

The formula (9) can be easily adapted to a parallel computation, using more than one processor. So, the following theorem holds:

Theorem. *The execution time needed to get the amount of money a person has to pay after n years is $O(\log n)$ using the double recursive technique on a binary tree communication among processors.*

Proof. Relation (9) can be written

$$\begin{bmatrix} a_{x+1} \\ -1 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{p(x, x+1)v} & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_x \\ -1 \end{bmatrix} \tag{10}$$

Analogous a_{x+2} depends on a_{x+1}, etc. Finally, in order to compute a_{x+n}, we have to compute only the matrices product which, on a binary tree connectivity, can be made in $O(\log n)$ (see [2]). So the theorem is proved.

References

[1] Blaga, P., Mureşan, A.S., Lupaş, A., *Matematici financiare şi actuariale*, Ed. Constant, Sibiu, 2001.

[2] Chiorean, I., *Calcul paralel*, Ed. Albastră, 1999.

[3] Mihoc, Gh., *Tratat de matematici actuariale*, Univ. Bucureşti, pp. 106-129.

[4] Purcaru, I., *Matematică şi asigurări*, Ed. Economică, Bucureşti, 1994.

Babeş-Bolyai University, Department of Mathematics and Informatics, Kogălniceanu, 1, Cluj-Napoca, Romania