Journal of Computational and Applied Mathematics 49 (1993) 207–216
North-Holland



CAM 1435

# Computing the coefficients of a recurrence formula for numerical integration by moments and modified moments

M. Morandi Cecchi
*Dipartimento di Matematica Pura ed Applicata, University of Padova, Italy*

M. Redivo Zaglia
*Dipartimento di Elettronica e Informatica, University of Padova, Italy*

Received 20 January 1992
Revised 22 April 1992

*Abstract*

Morandi Cecchi, M. and M. Redivo Zaglia, Computing the coefficients of a recurrence formula for numerical integration by moments and modified moments, Journal of Computational and Applied Mathematics 49 (1993) 207–216.

To evaluate the class of integrals $\int_{-1}^{1} e^{-\alpha x} f(x)\, dx$, where $\alpha \in \mathbb{R}^+$ and the function $f(x)$ is known only approximately in a tabular form, we wish to use a Gaussian quadrature formula. Nodes and weights have to be computed using the family of monic orthogonal polynomials, with respect to the weight function $e^{-\alpha x}$, obtained through the three-term recurrence relation $P_{k+1}(x) = (x + B_{k+1})P_k(x) - C_{k+1}P_{k-1}(x)$.

To guarantee a good precision, we must evaluate carefully the values for the coefficients $B_{k+1}$ and $C_{k+1}$. Such evaluations are made completely formally through a *Mathematica* program to obtain great precision.

A comparison between various methods, starting from moments and modified moments, is shown. Numerical results are also presented.

*Keywords:* Orthogonal polynomials; recurrence relation for orthogonal polynomials; Gaussian quadrature; moments; modified moments; symbolic computation

## 1. Introduction

In quantum mechanics it is very important to evaluate, with great precision, the class of integrals

$$\int_{-1}^{1} e^{-\alpha x} f(x)\, dx,$$

*Correspondence to:* Prof. M. Morandi Cecchi, Dipartimento di Matematica Pura ed Applicata, University of Padova, Via Belzoni 7, 35131 Padova, Italy.


*SSDI* 0377-0427(93)E0062-Q

with $\alpha \in \mathbb{R}^+$ and $f(x)$ known in a tabular form. Having a positive weight $e^{-\alpha x}$ and $\int_{-1}^{1} e^{-\alpha x}\,dx < +\infty$, we can use a Gaussian quadrature formula like

$$I = \int_{-1}^{1} e^{-\alpha x} f(x)\,dx = \sum_{i=1}^{n} A_i^{(n)} f(x_i) + R_n = I_n + R_n,$$

where $x_i$, $-1 < x_1 < \cdots < x_n < 1$, are the real zeros of the polynomial $P_n(x) \in \mathbb{P}_n$, orthogonal in $[-1, 1]$ with respect to $e^{-\alpha x}$ and the $A_i^{(n)}$'s are the weights, depending on $n$ and on the knots distribution. In such a way the formula is exact over $\mathbb{P}_{2n-1}$.

To determine the zeros and the weights, we must compute, with great accuracy, the coefficients of the polynomials $P_k \in \mathbb{P}_k$, satisfying the orthogonality conditions

$$c\big(x^i P_k(x)\big) = 0, \quad i = 0,\ldots,k-1,$$

where $c$ is a linear functional over the space $\mathbb{P}$ of real polynomials [2, Chapter 2] defined as

$$c(x^i) = c_i = \int_{-1}^{1} x^i e^{-\alpha x}\,dx.$$

Thus, after computing the first $2k$ moments $c_i$, the orthogonal polynomial $P_k(x) = a_0 + a_1 x + \cdots + a_k x^k$ can be determined by setting $a_k = 1$ and solving the classical system

$$\begin{cases} a_0 c_0 + a_1 c_1 + \cdots + a_{k-1} c_{k-1} = -c_k, \\ a_0 c_1 + a_1 c_2 + \cdots + a_{k-1} c_k = -c_{k+1}, \\ \vdots \\ a_0 c_{k-1} + a_1 c_k + \cdots + a_{k-1} c_{2k-2} = -c_{2k-1}. \end{cases}$$

This problem has been already considered in [12], where, for some significant values of $\alpha$ and $n$, zeros and weights have been provided. However, this system, always nonsingular, is often ill-conditioned. Thus, to avoid the computation of the $k+1$ coefficients $a_i$'s solving this system, we can determine them using the three-term recurrence relation

$$P_{-1}(x) = 0, \qquad P_0(x) = 1,$$
$$P_{k+1}(x) = (x + B_{k+1})P_k(x) - C_{k+1}P_{k-1}(x), \quad k = 0, 1, \ldots, \tag{1}$$

that is, computing the $2k$ coefficients $B_i$, $C_i$, for $i = 1,\ldots,k$.

Using this approach, a recursive algorithm to determine, for any value of $\alpha$ and $n$, the zeros and the weights has been proposed in [13]. This algorithm allows, in a very easy way, to increase the number of knots of a Gaussian quadrature formula using the values computed in the preceding formula. This is possible because only the moments, the coefficients of the two preceding orthogonal polynomials and the last computed zeros are needed.

As previously said, to get a good approximation of the integral, the coefficients of the polynomial can be determined in the best possible way. But, in all methods considered, the crucial point seems to be the computation of moments $c_i$ and that of modified moments $v_i$ (see Section 3).

Therefore, in this work comparisons among different methods based on moments and on modified moments are presented. We use the usual numerical approach but, above all, we consider the formal evaluation made through a *Mathematica* program. Such comparisons will show, in a formal way, the possible advantages of using modified moments methods, but also report some possible unpredictable results. All the comparisons presented here are made over the symbolic and numerical computation of the moments, of the modified moments and of the coefficients $B_i$ and $C_i$ of the recurrence formula (1).

## 2. Moments-based methods

Two moments-based methods have been considered in this paper. The first one, called *Chebyshev method* [5], allows to determine, starting from the moments, the coefficients of the recurrence relation. Unfortunately, this method is often ill-conditioned in practice.

Let us give a matrix $Z$ whose elements are defined as

$$z_{k,i}=c(P_k x^i), \quad k, i=0, 1, 2,\ldots,$$

where the $z_{k,i}$, for $i<k$, due to the orthogonality conditions, are always zero. The first row of this matrix represents the moments, since $z_{0,i}=c(P_0 x^i)=c(x^i)=c_i$.

Multiplying (1) by $x^i$ and applying the functional $c$, we obtain the Chebyshev method:

$$z_{k,i}=z_{k-1,i+1}+B_k z_{k-1,i}-C_k z_{k-2,i}, \quad i=k, k+1,\ldots,$$

$$B_{k+1}=\frac{z_{k-1,k}}{z_{k-1,k-1}}-\frac{z_{k,k+1}}{z_{k,k}}, \qquad C_{k+1}=\frac{z_{k,k}}{z_{k-1,k-1}},$$

for $k=1, 2,\ldots$ and with the initial conditions

$$z_{-1,i}=0, \quad i=1, 2,\ldots, \qquad z_{0,i}=c_i, \quad i=0, 1, 2,\ldots,$$

$$B_1=-\frac{c_1}{c_0}, \qquad C_1=c_0.$$

The second method, used in [13], directly gives the coefficients of the recurrence relation and also the coefficients of the orthogonal polynomial $P_{k+1}$. Thus no additional multiplications as in (1) have to be made.

Let us write $P_k(x)=a_0+a_1x+\cdots+a_kx^k$ as $P_k(x)=\sum_{i=0}^k p_i^{(k)}x^i=p_0^{(k)}+p_1^{(k)}x+\cdots+p_k^{(k)}x^k$ with $p_k^{(k)}=1$, $\forall k$. As shown in [2], the following relations hold:

$$B_{k+1}=-\frac{\alpha_k}{h_k}, \qquad C_{k+1}=\frac{h_k}{h_{k-1}},$$

where $h_k=c(x^kP_k)$ and $\alpha_k=c(x^{k+1}P_k)+p_{k-1}^{(k)}c(x^kP_k)$.

Putting them into the recurrence relation (1) and considering the coefficients of the powers of $x$, the coefficients $p_i^{(k)}$ can be computed recursively as follows [2]:

$$p_{-1}^{(-1)}=p_0^{(-1)}=0, \quad h_{-1}=1,$$

$$p_{-1}^{(0)}=p_1^{(0)}=0, \qquad p_0^{(0)}=1,$$

$$p_{-1}^{(k+1)}=p_{k+2}^{(k+1)}=0, \qquad p_{k+1}^{(k+1)}=1,$$

$$p_i^{(k+1)}=p_{i-1}^{(k)}+B_{k+1}p_i^{(k)}-C_{k+1}p_i^{(k-1)}, \quad i=0,\ldots,k,$$

where for $k=0, 1, \ldots$ the quantities

$$h_k = \sum_{i=0}^{k} c_{k+i} p_i^{(k)}, \qquad \gamma_k = \sum_{i=0}^{k} c_{k+i+1} p_i^{(k)}, \qquad \alpha_k = \gamma_k + p_{k-1}^{(k)} h_k,$$

$$B_{k+1} = -\frac{\alpha_k}{h_k}, \qquad C_{k+1} = \frac{h_k}{h_{k-1}}$$

have to be computed.

This recursive method can be viewed also as an application of the *bordering method* (see [6,15]). It permits to compute, starting from the solution of the initial system, in a recursive form, all the other solutions. In fact, the system to be solved for computing the coefficients $p_i^{(k)}$ can be obtained by adding a new equation and a new unknown to the preceding system used for computing the coefficients $p_i^{(k-1)}$. Thus the new matrix has been bordered by a new row and a new column. The bordering method works on a general linear system, and a version to avoid division by zero, called *block bordering method*, can be found in [3] with two subroutines corresponding to these methods. For determining the coefficients of the orthogonal polynomials, the matrices are always Hankel matrices and then the bordering method simplifies (see [16]), and finally we obtain the previous form of the method.

## 3. Modified moments-based methods

If the methods are based on the computation of moments, they may be severely ill-conditioned, as shown in [7].

In [14], using families of classical orthogonal polynomials (Legendre, Chebyshev,...), Sack and Donovan present a method, called *modified moments method*, for computing the coefficients of the recurrence relation. The sensitivity of orthogonal polynomials using this method has been studied in [10]. If the modified moments are accurately computed, the results proposed in various works (see, for example, [9,14]) seem to prove that the computation of the coefficients $B_i$ and $C_i$ becomes more stable.

In this approach, the moments are presented in a modified form like

$$v_i = c(\Pi_i), \quad i = 0, 1, \ldots,$$

where $\{\Pi_n\}$ is an auxiliary family of monic orthogonal polynomials satisfying

$$\Pi_{-1}(x) = 0, \qquad \Pi_0(x) = 1,$$

$$\Pi_{k+1}(x) = (x + B'_{k+1})\Pi_k(x) - C'_{k+1}\Pi_{k-1}(x), \quad k = 0, 1, \ldots .$$

Obviously, when $\Pi_i(x) = x^i$, then $v_i = c_i$.

When the intervals of the families $\{P_n\}$ and $\{\Pi_n\}$ are finite, the *modified Chebyshev algorithm* seems to be stable for computing the coefficients $B_i$ and $C_i$. This method, first proposed in [17] (see also [2,9]), is defined as follows.

We consider the matrix $Z$ with elements

$$z_{k,i} = c(P_k \Pi_i), \quad k, i = 0, 1, 2, \ldots .$$

All the $z_{k,i}$, with $i<k$, are zero, due to the orthogonality conditions, and the first row contains the modified moments because $z_{0,i}=c(P_0 \Pi_i)=c(\Pi_i)=v_i$.

The method, for $k=1, 2, \ldots$, is as follows:

$$z_{k,i}=z_{k-1,i+1}-B'_{i+1}z_{k-1,i}+C'_{i+1}z_{k-1,i-1}+B_k z_{k-1,i}-C_k z_{k-2,i}, \quad i=k,\ k+1,\ldots,$$

$$B_{k+1}=B'_{k+1}+\frac{z_{k-1,k}}{z_{k-1,k-1}}-\frac{z_{k,k+1}}{z_{k,k}}, \qquad C_{k+1}=\frac{z_{k,k}}{z_{k-1,k-1}},$$

with the initial conditions

$$z_{-1,i}=0, \quad i=1, 2,\ldots, \qquad z_{0,i}=v_i, \quad i=0, 1, 2,\ldots,$$

$$B_1=B'_1-\frac{v_1}{v_0}, \qquad C_1=v_0.$$

If we choose $\Pi_i=x^i$, this method reduces to the Chebyshev method described in Section 2.

As shown in [4], it is not necessary that the family $\{\Pi_n\}$ be orthogonal, and the modified moments can be built up starting from any family of polynomials.

## 4. Computing moments and modified moments

For our integral, either moments and modified moments are analytically computable. For computing the moments, two formulas have been proposed in [13] and the following theorems hold.

Let $c_i=\int_{-1}^{1}x^i e^{-\alpha x}\,dx$, $\alpha\in\mathbb{R}$, $i=0, 1,\ldots$. We proved the following theorems.

**Theorem 4.1.**

$$c_0=-\frac{1}{\alpha}(e^{-\alpha}-e^{\alpha}), \qquad \alpha c_i=-\left[e^{-\alpha}-(-1)^i e^{\alpha}\right]+ic_{i-1}, \quad i=1, 2,\ldots .$$

**Theorem 4.2.**

$$c_i=(-1)^i i!\sum_{j=0}^{i}\frac{(-1)^j}{\alpha^{j+1}(i-j)!}\left[e^{\alpha}-(-1)^{i+j}e^{-\alpha}\right].$$

For modified moments, we choose the family of monic orthogonal polynomials

$$\Pi_0(x)=T_0(x), \qquad \Pi_n(x)=\frac{T_n(x)}{2^{n-1}}, \quad n=1, 2,\ldots,$$

where the $T_n$'s are the Chebyshev polynomials of the first kind. This choice seems to be suitable because of the coincidence of the interval of definition of the two families. The auxiliary polynomials are defined by the following relations:

$$\Pi_{-1}(x)=0, \qquad \Pi_0(x)=T_0(x)=1,$$

$$\Pi_{k+1}(x)=x\Pi_k(x)-C'_{k+1}\Pi_{k-1}(x), \quad k=0, 1,\ldots,$$

$$B'_{k+1}=0, \quad k=0, 1,\ldots, \qquad C'_{k+1}=\tfrac{1}{2}, \quad k=0, 1, \qquad C'_{k+1}=\tfrac{1}{4}, \quad k=2, 3,\ldots .$$

For computing the modified moments, two theorems directly obtained from Theorems 4.1 and 4.2 hold.

We consider the modified moments $v_i = c(\Pi_i) = \int_{-1}^{1} \Pi_i(x) e^{-\alpha x}\, dx$, $\alpha \in \mathbb{R}$, $i = 0, 1, \ldots$, where $\Pi_i$ is the $i$th Chebyshev monic polynomial of the first kind. Thus we have the following theorem.

**Theorem 4.3.**

$$v_0 = c_0, \qquad v_1 = c_1,$$

$$v_i = i \sum_{m=0}^{[i/2]} (-1)^m \frac{(i-m-1)!}{m!\,(i-2m)!} 2^{-2m} c_{i-2m}, \quad i = 2, 3, \ldots .$$

**Proof.** Having $\Pi_0(x) = 1$ and $\Pi_1(x) = x$, the first two relations immediately hold. The explicit form of $T_i(x)$ (see, for example, [1]) is as follows:

$$\begin{aligned} T_i(x) &= \tfrac{1}{2} i \sum_{m=0}^{[i/2]} (-1)^m \frac{(i-m-1)!}{m!\,(i-2m)!} (2x)^{i-2m} \\ &= i 2^{i-1} \sum_{m=0}^{[i/2]} (-1)^m \frac{(i-m-1)!}{m!\,(i-2m)!} 2^{-2m} x^{i-2m}. \end{aligned}$$

Thus,

$$\Pi_i(x) = i \sum_{m=0}^{[i/2]} (-1)^m \frac{(i-m-1)!}{m!\,(i-2m)!} 2^{-2m} x^{i-2m}.$$

Applying the functional $c$ to both sides of the relation, we obtain

$$\begin{aligned} v_i = c(\Pi_i(x)) &= i \sum_{m=0}^{[i/2]} (-1)^m \frac{(i-m-1)!}{m!\,(i-2m)!} 2^{-2m} c(x^{i-2m}) \\ &= i \sum_{m=0}^{[i/2]} (-1)^m \frac{(i-m-1)!}{m!\,(i-2m)!} 2^{-2m} c_{i-2m}. \qquad \square \end{aligned}$$

**Theorem 4.4.**

$$v_i = i \sum_{m=0}^{[i/2]} (-1)^{i-m} 2^{-2m} \frac{(i-m-1)!}{m!} \sum_{j=0}^{i-2m} \frac{(-1)^j}{\alpha^{j+1}(i-2m-j)!} \left[ e^{\alpha} - (-1)^{i-2m+j} e^{-\alpha} \right],$$

$$i = 2, 3, \ldots .$$

**Proof.** Substituting the formula of Theorem 4.2 into that of Theorem 4.3 gives the result after trivial simplifications. $\square$

## 5. Numerical results

The numerical results, obtained on a μVAX 3100 using H-float storage to obtain a good precision (about 33 significant decimal digits), do not permit to decide if, in our case, the

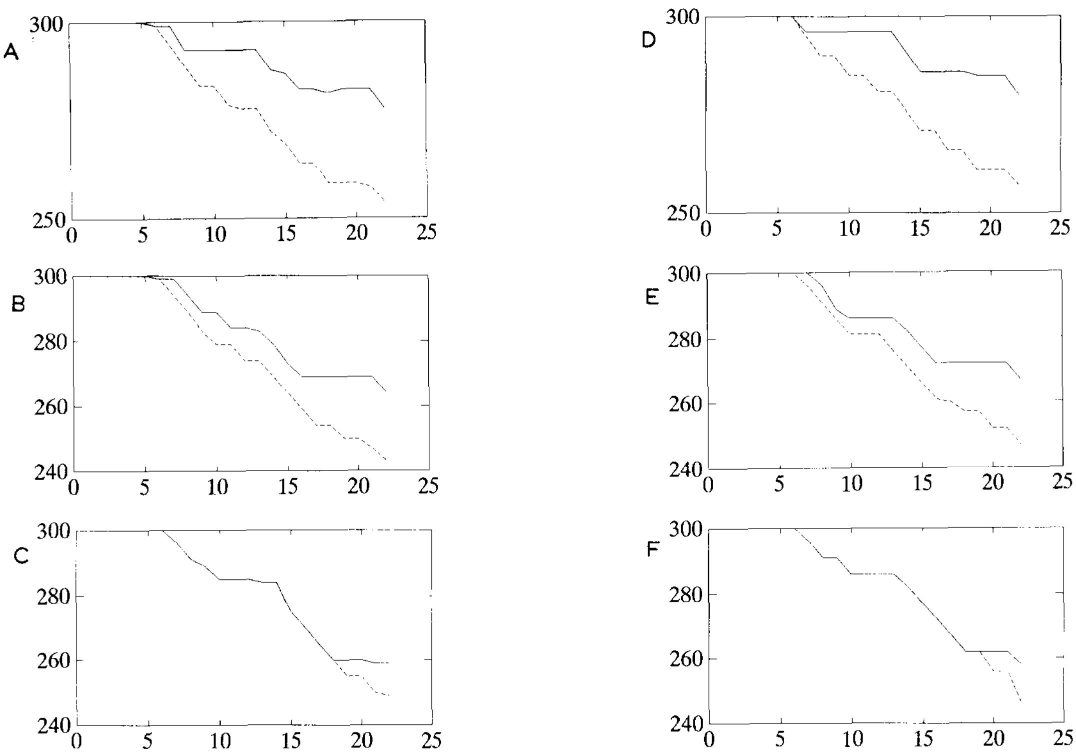


Fig. 1. (A) $B_i$, $\alpha = 2.0$; (B) $B_i$, $\alpha = 5.0$; (C) $B_i$, $\alpha = 15.0$; (D) $C_i$, $\alpha = 2.0$; (E) $C_i$, $\alpha = 5.0$; (F) $C_i$, $\alpha = 15.0$.

modified moments methods give a better computation of the coefficients $B_i$ and $C_i$. Thus we have decided to choose the symbolic computation in order to look for a correct answer to the question.

For performing this symbolic computation we considered *Mathematica* [18], running on a Macintosh II. This program allows to specify formal calculations requiring a certain number of exact decimal digits (in our case we have chosen 300 digits). When the operations of simplification introduce uncertainty over some digits, then the program automatically reduces this number and returns as output only the digits considered to be exact.

In all the comparisons presented here, three values for $\alpha$, namely $\alpha = 2.0$, 5.0 and 15.0, and a degree for the orthogonal polynomial $P_i$ varying from $i = 1$ to $i = 22$ have been considered.

First of all, in Fig. 1 we consider a comparison between the coefficients $B_i$ and $C_i$ formally obtained with *Mathematica*, using the Chebyshev method and the modified Chebyshev method. In the $x$-axis the values of $i$ are reported. In the $y$-axis the number of decimal digits given in output by *Mathematica* (and then considered exact) are represented. As we can see, the behaviour shows that the modified Chebyshev algorithm (solid lines) gives, for the values $\alpha = 2.0$ and $\alpha = 5.0$, better results than those of the Chebyshev method (dash lines). When $\alpha$ increases, the behaviour of the two methods becomes almost the same.

In Fig. 2 a comparison between the numerical results and the results obtained with *Mathematica* is shown. We denote with dash lines the number of common digits between the numerical results obtained with three numerical methods: Chebyshev method, bordering

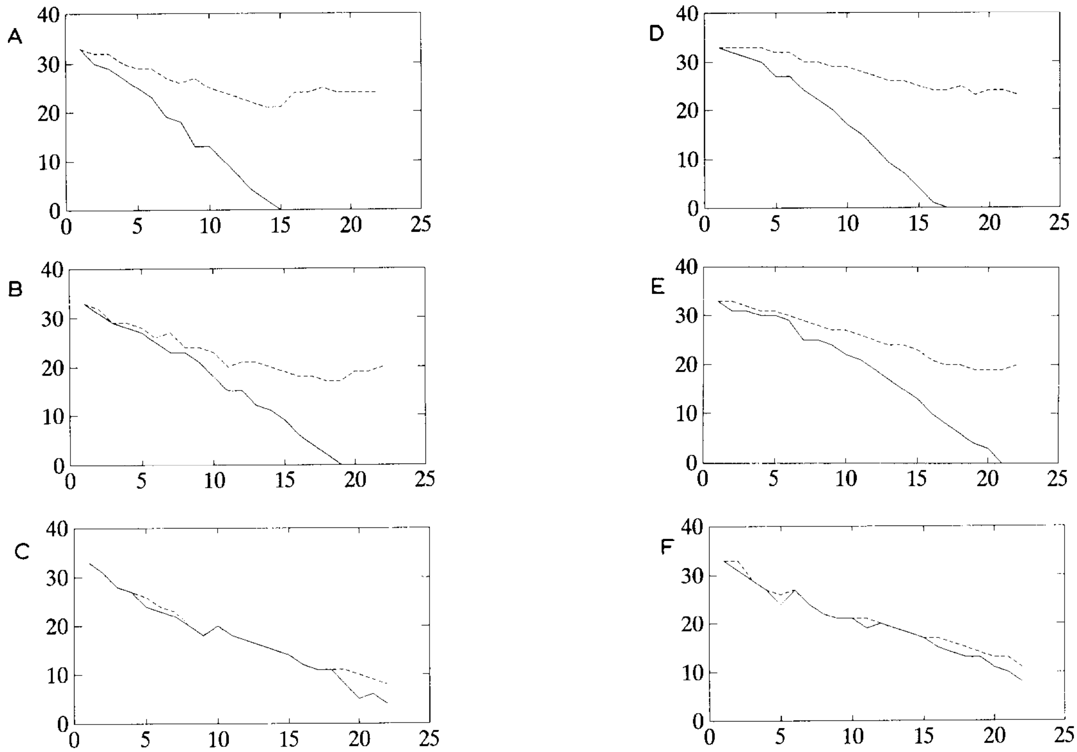


Fig. 2. (A) $B_i$, $\alpha = 2.0$; (B) $B_i$, $\alpha = 5.0$; (C) $B_i$, $\alpha = 15.0$; (D) $C_i$, $\alpha = 2.0$; (E) $C_i$, $\alpha = 5.0$; (F) $C_i$, $\alpha = 15.0$.

method and modified Chebyshev method. The solid lines represent the number of common digits between all the numerical results and the results obtained with *Mathematica*. The graphical representation clearly shows that even if one can believe that the common digits

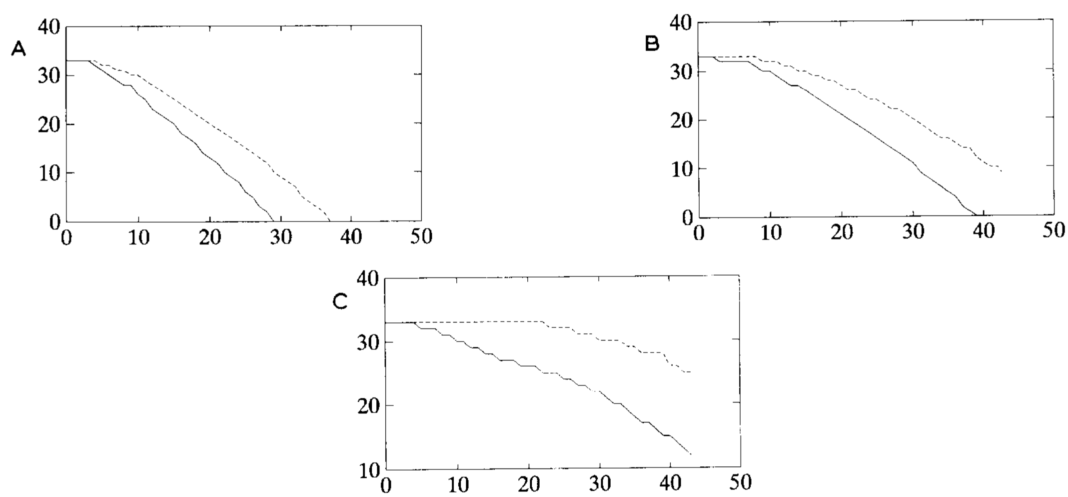


Fig. 3. (A) $\alpha = 2.0$; (B) $\alpha = 5.0$; (C) $\alpha = 15.0$.

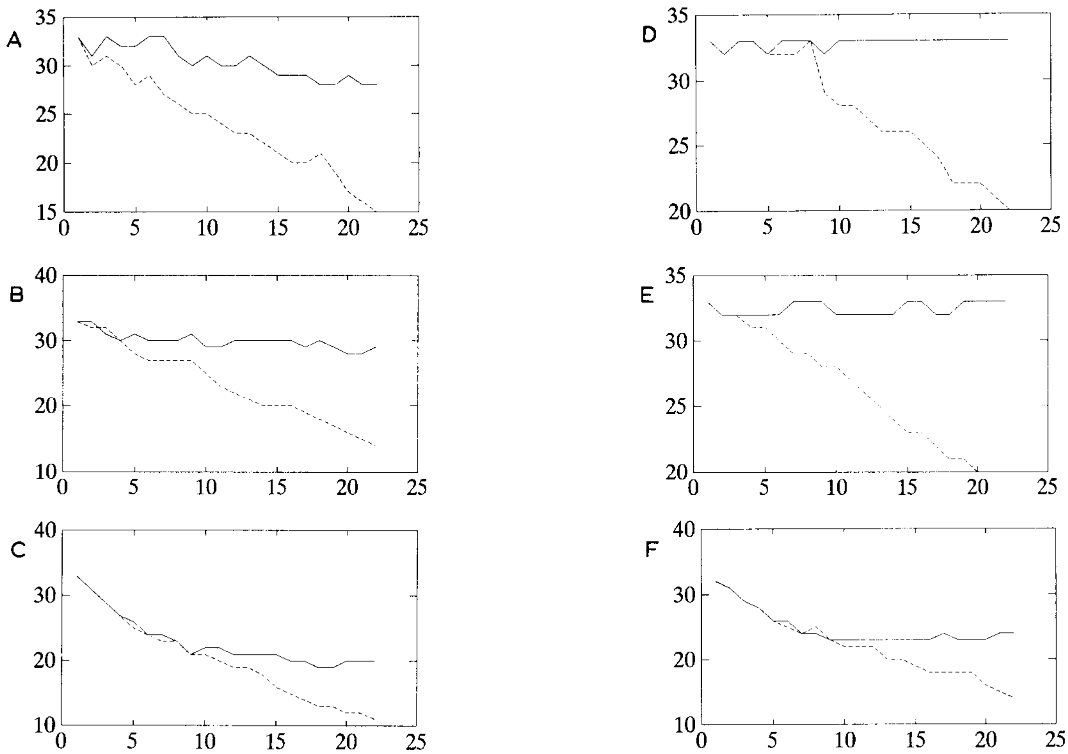


Fig. 4. (A) $B_i$, $\alpha = 2.0$; (B) $B_i$, $\alpha = 5.0$; (C) $B_i$, $\alpha = 15.0$; (D) $C_i$, $\alpha = 2.0$; (E) $C_i$, $\alpha = 5.0$; (F) $C_i$, $\alpha = 15.0$.

between the three numerical methods are "exact" digits, in reality only a lower number of these digits are "exact". Thus, one can say that the three methods give a wrong answer with the same behaviour.

The unsolved problem is to understand why the modified moments method (certainly more stable for the results given in Fig. 1) loses its advantages mostly when $\alpha = 2.0$ and $\alpha = 5.0$. Figure 3 shows for moments (dash lines) and modified moments (solid lines), computed numerically with the analytical formulas, the number of common digits with the results obtained with *Mathematica*. As we can see, the computation of modified moments is less stable than that of the moments, and thus this is probably the reason for the results of Fig. 2.

To prove this hypothesis, we give to the numerical program using the Chebyshev method (dash lines) and the modified Chebyshev method (solid lines), the values of the moments and of the modified moments obtained by *Mathematica*, namely we give to the program the "exact" values for such quantities. In such a way we retrieve in Fig. 4 a behaviour similar to that of Fig. 1 and the results given by the modified moments method are really more stable.

## 6. Conclusions

Starting from our results, some remarks can be made. The computations made with the symbolic approach certify that, using modified moments, the methods for computing the coefficients of the recurrence relation are really more stable. A condition "near necessary" anyway is to have the modified moments in a closed form. But, also in that case, we have no guarantee that the modified moments increase the stability. In fact, the numerical computation of the modified moments (starting from our analytical relation) can be, in some cases, less stable than the computation of the moments and thus the results do not improve. It should be noted that the contribution of symbolic computation is essential in the understanding of the behaviour of the recurrence formula considered.

Various papers deal with the methods using moments and modified moments. Recently, Golub and Gutknecht [11], extending the theory to the case of an indefinite weight function, presented a review of the basic algorithms. On the problem of Gaussian quadrature with modified moments, one can see [8].

## References

[1] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions* (Dover, New York, 1965).
[2] C. Brezinski, *Padé-Type Approximation and General Orthogonal Polynomials*, Internat. Ser. Numer. Math. **50** (Birkhäuser, Basel, 1980).
[3] C. Brezinski and M. Redivo Zaglia, *Extrapolation Methods. Theory and Practice,* Stud. Comput. Math. **2** (North-Holland, Amsterdam, 1991).
[4] C. Brezinski and M. Redivo Zaglia, A new presentation of orthogonal polynomials with applications to their computation, *Numer. Algorithms* **1** (1991) 207–222.
[5] P.L. Chebyshev, Sur les fractions continues, *J. Math. Pures Appl. (2)* **3** (1858) 289–323.
[6] V.N. Faddeeva, *Computational Methods of Linear Algebra* (Dover, New York, 1959).
[7] W. Gautschi, Construction of Gauss–Christoffel quadrature formulas, *Math. Comp.* **22** (1968) 251–270.
[8] W. Gautschi, On the construction of Gaussian quadrature rules from modified moments, *Math. Comp.* **24** (1970) 245–260.
[9] W. Gautschi, On generating orthogonal polynomials, *SIAM J. Sci. Statist. Comput.* **3** (1982) 289–317.
[10] W. Gautschi, On the sensitivity of orthogonal polynomials to perturbation in the moments, *Numer. Math.* **48** (1986) 369–382.
[11] G.H. Golub and M.H. Gutknecht, Modified moments for indefinite weight functions, *Numer. Math.* **57** (1990) 607–624.
[12] M. Morandi Cecchi, L'integrazione numerica di una classe di integrali utili nei calcoli quantomeccanici, *Calcolo* **4** (3) (1967) 363–368.
[13] M. Morandi Cecchi and M. Redivo Zaglia, A new recursive algorithm for a Gaussian quadrature formula via orthogonal polynomials, in: C. Brezinski, L. Gori et al., Eds., *IMACS Transactions on Orthogonal Polynomials and their Applications* (Baltzer, Basel, 1991) 353–358.
[14] R.A. Sack and A.F. Donovan, An algorithm for Gaussian quadrature given modified moments, *Numer. Math.* **18** (1971/72) 465–478.
[15] J. Sherman and W.J. Morrison, Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix, *Ann. Math. Statist.* **20** (1949) 621.
[16] W.F. Trench, An algorithm for the inversion of finite Hankel matrices, *SIAM J.* **13** (1965) 1102–1107.
[17] J.C. Wheeler, Modified moments and Gaussian quadrature, *Rocky Mountain J. Math.* **4** (1974) 287–296.
[18] S. Wolfram, *Mathematica. A System for doing Mathematics by Computer* (Addison-Wesley, Redwood City, CA, 2nd ed., 1991).