JOURNAL OF COMPUTATIONAL PHYSICS 99, 321–336 (1992)

A Domain Decomposition Method for Generating Orthogonal Polynomials for a Gaussian Weight on a Finite Interval

RAYMOND C. Y. CHIN*

Lawrence Livermore National Laboratory, Livermore, California 94550

Received June 7, 1990; revised March 28, 1991

A domain decomposition method has been developed for generating orthogonal polynomials for a Gaussian weight on $(-1, 1)$. The method takes advantage of the underlying asymptotic structure of the orthogonal polynomials and, hence, it is *effective* in the sense that it makes maximal use of the analytic properties of the solution to increase accuracy and efficiency. These polynomials are necessary for constructing Gaussian quadrature formulas that are encountered in large quantum chemistry computational packages and in calculating the Compton scattering kernel and its associated angular moments.

I. INTRODUCTION

This paper is concerned with the construction of Gaussian quadrature given a positive weight function or a measure on a finite interval. There are two steps in such a construction: first, the associated orthogonal polynomials must be generated; and, second, the Gaussian quadrature weights and abscissas are calculated as an eigenvalue problem associated with the coefficient matrix corresponding to the three-term recurrence satisfied by the orthogonal polynomials. The second step is the less difficult of the two because the Francis QR algorithm is an effective tool. On the other hand, the problem of generating orthogonal polynomials given just a weight is fraught with numerical difficulties.

In theory, the *procedure of Stieltjes* consists of applying the Gram–Schmidt orthogonalization process to a subspace of polynomials. The sequence of orthogonal polynomials then satisfies a three-term recurrence whose coefficients are just the ratios of the weighted norms of successive polynomials and the weighted centroid of the polynomials.

In practice, the calculations involve integrating a $2n$th-degree polynomial for each nth-order coefficient. This process is extremely delicate as is the solution of the three-term recurrence. Thus, numerical stability is a well-documented concern [1]. A relatively effective algorithm for generating the recurrence coefficients is the "modified moments" method. Basically, the "modified moments" method makes use of a known and suitable system of polynomials that is "close" in some sense to the desired polynomials to precondition the calculation. This process has two steps; one step involves changing of bases between polynomials subspaces and the other step is concerned with preconditioning the computation. The role played by changing bases is to develop a relation between the coefficients of the respective recurrences and the matrix elements of the linear map. The matrix elements are obtainable from the "modified moments" and, thus, enabling the computation of the desired recurrence coefficients in terms of known quantities. The role of preconditioning is to enhance numerical stability. As we shall see later, the "modified moments" can be interpreted as the "Fourier" coefficients of the ratio of the actual weight to that of the weight of the preconditioning orthogonal polynomial subspace to which the projection is performed. So far, only the classical orthogonal polynomials have been used as preconditioners, since they are well known otherwise we face the problem of generating orthogonal polynomials, which is the topic of this paper. Just what constitutes an effective preconditioner is an open question.

For the present case of a Gaussian or a Hermite weight function, integrals involving the weight

$$\int_{-1}^{1} f(x) \exp(-b^2 x^2)\, dx$$

are commonly encountered in quantum chemistry calculations. They are integral parts of large quantum chemistry packages such as the GAUSSIANXX series [2], the COLUMBUS MRCI Program System [3], HONDO [4], etc. A relatively large portion of the computation resides in the evaluation of these integrals. As a compromise to efficiency and storage, the developers of these packages

* Now at Indiana University-Purdue University at Indianapolis, Department of Computer and Information Science, 723 West Michigan Street, Indianapolis, IN 46202-5132.

0021-9991/92 $3.00

store the quadrature weights and abscissas in memory. A fast algorithm for generating the Gaussian quadrature formulae, if it exists, will undoubtedly alleviate such storage requirements and, thus, free the user to pursue other pressing tasks requiring larger blocks of storage. Another application of the quadrature formulas is in computing a Compton scattering kernel and its associated angular moments [5]. Compton scattering from relativistic electrons plays a primary role in the theory of radiative transport.

In a recent survey, Gautschi [6] lists Piessens and Branders [7] and King and Dupuis [8] as the main contributors to the computation of the Gaussian quadrature rule for the Gaussian weight on the interval $(-1, 1)$. Piessens and Branders use the "modified moments" method with the Chebyshev polynomials of the first kind as the preconditioner. They calculated the Gaussian quadrature rule for only a limited range of values of b and n, the order of the polynomial. In constrast, King and Dupuis are only interested in a restricted set of parameters, $n \leqslant 10$ and any b. They also give an analysis of the local behaviors of the constructed polynomials, called Rys polynomials in a tribute to a colleague. King and Dupuis note in particular that as b increases the Rys polynomials approach the behavior of the scaled Hermite polynomials for $n \leqslant N$, where "N is such that b^2 is roughly larger than three times the largest zero of a $2N$ Hermite polynomial." This is a significant observation, namely, that the Rys polynomials can acquire certain behaviors in the parameter space of b and n.

The object of this paper is to take advantage of King and Dupuis' observation to develop an *effective* algorithm for generating Rys polynomials for all n and b using a domain decomposition method. It is *effective* in the sense that the algorithm makes maximal use of the analytic properties of the solution to increase efficiency and accuracy. This is also true of domain decomposition methods in solving partial differential equations with many and differing local substructures. In this paper, we will identify other sub-regions of the changing solution structure by employing techniques of asymptotic analysis. This is done by comparing the scale of variation of the Gaussian weight, b, and that of the scales inherent in the oscillation of the polynomials. Another objective is to investigate the effectiveness of various preconditioners in the "modified moments" method in light of these identifiable and distinguished local structures.

These inherent distinguished local behaviors match well with the basic tenets of the domain decomposition methods [9–11] and, therefore, can be fully and fruitfully exploited. The essence of this class of methods is the patching of differing local sub-domain solutions to form a global solution. Another laudable feature of the domain decomposition methods is that in each subdomain we may use an entirely different solution technique. This allows for increased efficiency and accuracy. These are the reasons underlying their popularity in treating problems with many and differing local structures, i.e., multiple scales problems. It is reasonable to expect that domain decomposition methods are suitable for generating orthogonal polynomials given any positive weight.

The organization of the paper is as follows. Section II defines the problem of generating orthogonal polynomials given just the weight. Section III reviews the "modified moments" method. Both Sections II and III are included for completeness and for the presentation to be self-contained. Section IV is concerned with considerations leading to the development of a domain decomposition method. Section V gives the algorithm. Section VI closes with results of the computation and discussions on effectiveness of the preconditioning polynomial subspaces.

II. GENERATION OF ORTHOGONAL POLYNOMIALS

This section reviews briefly the problem of generating orthogonal polynomials given a positive weight on a finite interval. The subsequent calculation of the Gaussian quadrature rule will also be discussed. There is no new information reported.

The *procedure of Stieltjes* for generating orthogonal polynomials given a positive weight, $w(x)$ on a finite interval (x_0, x_1) may be stated as follows. Let $\{\Pi_n\}$ be a set of monic orthogonal polynomials defined on the interval (x_0, x_1). Then they satisfy a three-term recurrence relation given by

$$\Pi_{n+1}(x) = (x - \alpha_n)\,\Pi_n(x) - \beta_n \Pi_{n-1}(x) \tag{1}$$

with

$$\Pi_{-1}(x) = 0 \qquad \text{and} \qquad \Pi_0(x) = 1. \tag{2}$$

Orthogonality imposes, as in the Gram–Schmidt process, a relation between the coefficients α_n and β_n and the integrals of the polynomials:

$$\gamma_n = \int_{x_0}^{x_1} \Pi_n^2(x)\, w(x)\, dx, \qquad n = 0, 1, 2, \ldots \tag{3}$$

$$\alpha_n = \gamma_n^{-1} \int_{x_0}^{x_1} x \Pi_n^2(x)\, w(x)\, dx, \qquad n = 0, 1, 2, \ldots \tag{4}$$

and

$$\beta_n = \gamma_n / \gamma_{n-1}, \qquad n = 1, 2, \ldots. \tag{5}$$

Note that γ_n is just the weighted norm of $\Pi_n(x)$; and α_n and β_n are respectively the weighted centroid of the polynomial and the ratio of successive weighted norms. Simply stated,

the problem of generating orthogonal polynomials given a weight, $w(x)$ on (x_0, x_1) is to compute the recurrence coefficients, α_n and β_n. We remark that the integrands of (3)–(4) are essentially polynomials of degree $2n$ and $2n+1$ and, therefore, their numerical integration can be delicate especially for large n.

The computational process of the *Stieltjes procedure* begins with (2), (3), and (4), yielding γ_0 and α_0. Next, substituting (2) into (1) gives $\Pi_1(x)$. Continuing in this manner, we can generate as many polynomials as desired. We remark that this is a *forward marching* process. It is well known that calculating three-term recurrence by forward marching can be troublesome due to numerical stability. Coupling this with the delicacy of the numerical integration, it is not unexpected that the computation is extremely ill-conditioned.

Having thus obtained the recurrence coefficients, α_n, $n=0, 1, ..., k$, and β_n, $n=1, ..., k$, we can form a symmetric, $k \times k$, tridiagonal recurrence coefficient matrix:

$$\begin{pmatrix} \alpha_0 & \sqrt{\beta_1} & & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ & & & \sqrt{\beta_{k-2}} & \alpha_{k-2} & \sqrt{\beta_{k-1}} \\ & & & & \sqrt{\beta_{k-1}} & \alpha_{k-1} \end{pmatrix}. \quad (6)$$

The eigenvalues of the above Jacobi matrix are the abscissas of the Gaussian quadrature rule, since this is equivalent to finding the zeroes of the kth polynomial, $\Pi_k(x)=0$. The corresponding Gaussian quadrature weights are related to the first component of the corresponding normalized eigenvectors. This computation can be accomplished effectively using Francis' QR algorithm [12].

For a symmetric weight function, $w(-x)=w(x)$ on a symmetric interval about the origin, $(-1, 1)$ as in the present case, the system of orthogonal polynomials simplifies to two sub-systems with even and odd parity. An immediate consequence is that

$$\alpha_n \equiv 0, \forall n,$$

and (1) becomes

$$\Pi_{n+1}(x) = x\Pi_n(x) - \beta_n \Pi_{n-1}(x). \quad (7)$$

The odd and even parity sub-spaces satisfy a three-term recurrence similar to (1): Let

$$\Pi_{2n}(x) \equiv Q_n(x) \quad \text{and} \quad \Pi_{2n+1}(x) \equiv xR_n(x); \quad (8)$$

then

$$Q_{n+1}(x) = [x^2 - (\beta_{2n} + \beta_{2n+1})] \times Q_n(x) - \beta_{2n}\beta_{2n-1} Q_{n-1}(x), \quad (9)$$

with

$$Q_{-1}(x) = 0 \quad \text{and} \quad Q_0(x) = 1, \quad (10)$$

and

$$R_{n+1}(x) = [x^2 - (\beta_{2n+1} + \beta_{2n+2})] \times R_n(x) - \beta_{2n}\beta_{2n+1} R_{n-1}(x), \quad (11)$$

with

$$R_{-1}(x) = 0 \quad \text{and} \quad R_0(x) = 1. \quad (12)$$

Substituting (8a), (8b) into (3), we obtain

$$\gamma_{2n} = \int_0^1 Q_n^2(x^2) \frac{w(x^2)}{\sqrt{x^2}} d(x^2) \quad (13)$$

and

$$\gamma_{2n+1} = \int_0^1 R_n^2(x^2)\, w(x^2) \sqrt{x^2}\, d(x^2). \quad (14)$$

Thus, we see from (13) and (14) that $Q_n(x^2)$ and $R_n(x^2)$ are respectively systems of half-range polynomials orthogonal with respect to the weights $w(x^2)/\sqrt{x^2}$ and $w(x^2)\sqrt{x^2}$ in the interval $0 \leqslant \eta \leqslant 1$, in which we may replace x^2 by η.

III. THE "MODIFIED MOMENTS" METHOD

III.1. *Theoretical Basis*

The "modified moment" or the modified Chebyshev's method is a relatively stable method for generating orthogonal polynomial on a finite interval. Although the derivation of the "modified moments" method can be found elsewhere, it is included for the sake of completeness and for making connection with the analysis leading to the development of the domain decomposition method. As we shall see, the basis of the method is concerned with preconditioning and with changing bases. The main idea is to precondition the calculation with a known orthogonal polynomial subspace of equal dimension. Stabilization is obtained if the preconditioned polynomial subspace is "close" in some sense to the actual orthogonal polynomial subspace. The two steps of the procedure, namely, changing basis and preconditioning the calculation are totally independent.

The process of changing bases is given by a linear transformation T

$$T: \mathscr{P}_w \to \mathscr{P}_{\hat{w}}$$

between two orthogonal polynomial subspaces $\mathscr{P}_w$ and $\mathscr{P}_{\hat{w}}$, of which $\mathscr{P}_{\hat{w}}$ is presumed known. The orthogonal polynomial subspace $\mathscr{P}_w$ is the set of monic polynomials orthogonal on the interval (x_0, x_1) with respect to the weight w and is defined by (1)–(5).

The orthogonal polynomial subspace $\mathscr{P}_{\hat{w}}$ is a known set of polynomials, not necessarily monic, orthogonal on an equivalent finite interval with respect to the weight $\hat{w}$, and is defined by the recurrence

$$\hat{P}_{n+1}(x) = (a_n x + b_n)\,\hat{P}_n(x) - c_n \hat{P}_{n-1}(x)$$

with

$$\hat{P}_{-1}(x) = 0 \qquad \text{and} \qquad \hat{P}_0(x) = 1.$$

Here, the coefficients a_n, b_n, and c_n are presumed known.

From the definitions of the polynomial subspaces, it is clear that the linear transformation is finite dimensional and the associated $N \times N$ matrix is lower triangular. Let the elements of the transformation matrix T be $t_{k,l}$, and

$$\hat{P}_n(x) = \sum_{l=0}^{n} t_{n,l} \Pi_l(x), \qquad n = 0, 1, \ldots, N-1; \tag{15}$$

then, the matrix elements $t_{k,l}$ may be computed recursively as

$$\begin{aligned} &t_{0,0} = 1; \\ &t_{1,0} = (a_0\alpha_0 + b_0)\,t_{0,0}, \qquad t_{1,1} = a_0 t_{0,0}; \end{aligned} \tag{16}$$

for $n \geqslant 2$;

$$t_{n,n} = a_{n-1} t_{n-1,n-1}, \tag{17}$$

$$\begin{aligned} t_{n,n-1} &= (a_{n-1}\alpha_{n-1} + b_{n-1})\,t_{n-1,n-1} \\ &\quad + a_{n-1} t_{n-1,n-2}, \end{aligned} \tag{18}$$

$$\begin{aligned} t_{n,m} &= a_{n-1} t_{n-1,m-1} \\ &\quad + (a_{n-1}\alpha_m + b_{n-1})\,t_{n-1,m} \\ &\quad + a_{n-1}\beta_{m+1} t_{n-1,m+1} - c_{n-1} t_{n-2,m}, \end{aligned} \tag{19}$$

with $m = 1, \ldots, n-2$ and

$$\begin{aligned} t_{n,0} &= (a_{n-1}\alpha_0 + b_{n-1})\,t_{n-1,0} \\ &\quad + a_{n-1}\beta_1 t_{n-1,1} - c_{n-1} t_{n-2,0}. \end{aligned} \tag{20}$$

Note that (16)–(20) involve both sets of recurrence coefficients, $\{\alpha_n, \beta_n\}$ and $\{a_n, b_n, c_n\}$, as well as the matrix elements $t_{k,l}$ for $1 \leqslant l \leqslant n$ and $k < n$.

The "modified moments" or the modified Chebyshev's method is related to the above change of bases algorithm through the introduction of the "mixed moments":

$$\begin{aligned} \sigma_{n,m} &= \int_a^b \Pi_n(x)\,\hat{P}_m(x)\,w(x)\,dx, \\ &\qquad n, m = 0, 1, 2, \ldots, N-1. \end{aligned} \tag{21}$$

The meaning of the "mixed moments" matrix element $\sigma_{n,m}$ will become evident as we substitute (15) into (21) to yield

$$\begin{aligned} \sigma_{n,m} &= \int_{x_0}^{x_1} w(x)\,dx\,\Pi_n(x) \sum_{l=0}^{m} t_{m,l}\Pi_l(x) \\ &= \sum_{l=0}^{m} t_{m,l} \int_{x_0}^{x_1} w(x)\,dx\,\Pi_n(x)\,\Pi_l(x). \end{aligned}$$

By virtue of orthogonality among the polynomials, $P_n(x)$ on (x_0, x_1), we have

$$\begin{aligned} \sigma_{n,m} &= \sum_{l=0}^{m} t_{m,l}\delta_{n,l}\gamma_n \\ &= \gamma_n t_{m,n}. \end{aligned} \tag{22}$$

Hence, we see that the "modified moments" method is intimately related to changing bases between polynomial subspaces.

The "modified moments" algorithm is the result of substituting (22) into (16)–(20) thus allowing the coefficients, α_n, $n = 0, 1, \ldots, N-1$, and β_n, $n = 1, 2, \ldots, N-1$, to be computed in terms of the "modified moments", $\sigma_{0,n}$, $n = 0, 1, 2, \ldots, 2N-1$. Explicit representations of α_n and β_n are derived by substituting (22) into (17) and (18), respectively, to arrive at

$$\beta_n = \frac{\sigma_{n,n}}{a_{n-1}\sigma_{n-1,n-1}}, \qquad 1 \geqslant n \geqslant N-1, \tag{23}$$

and

$$\begin{aligned} \alpha_n &= \frac{\sigma_{n,n+1}}{a_n \sigma_{n,n}} - \frac{\sigma_{n-1,n}}{a_{n-1}\sigma_{n-1,n-1}} - \frac{b_n}{a_n}, \\ &\qquad 1 \geqslant n \geqslant N-1. \end{aligned} \tag{24}$$

For $n = 0$, we have $\beta_0 = 0$ (here, β_0 is arbitrary and is set equal to 0) and (16a) becomes

$$\alpha_0 = \frac{\sigma_{0,1}}{a_0\sigma_{0,0}} - \frac{b_0}{a_0}.$$

To compute $\sigma_{n-1,n}$ and $\sigma_{n,n}$, we begin with the calculation of the $2N$ "modified moments,"

$$\sigma_{0,n}=\int_a^b \hat{P}_n(x)\, w(x)\, dx, \qquad n=0, 1, 2, ..., 2N-1. \tag{25}$$

The choice of the preconditioning polynomial subspace is explicitly made here.

For $m=1$, we obtain from (20) and (22)

$$\sigma_{1,n}=\frac{1}{a_n}\sigma_{0,n+1}-\left(\alpha_0+\frac{b_n}{a_n}\right)\sigma_{0,n} + \frac{c_n}{a_n}\sigma_{0,n-1}, \qquad 1\leqslant n\leqslant 2N-2. \tag{26}$$

For $m=2, ..., N$ and $m\leqslant n\leqslant 2N-1-m$, we have from (19)

$$\sigma_{m,n}=\frac{1}{a_n}\sigma_{m-1,n+1}-\beta_{m-1}\sigma_{m-2,n} -\left(\alpha_{m-1}+\frac{b_n}{a_n}\right)\sigma_{m-1,n}+\frac{c_n}{a_n}\sigma_{m-1,n-1}. \tag{27}$$

This completes the algorithm.

We note that the "modified moments" $\sigma_{0,n}$ can be rewritten as

$$\sigma_{0,n}=\int_{x_0}^{x_1}\hat{P}_n(x)\left\{\frac{w(x)}{\hat{w}(x)}\right\}\hat{w}(x)\, dx. \tag{28}$$

This permits us to interpret the "modified moment," $\sigma_{0,n}$, $n=0, 1, ..., 2N-1$, as the expansion or "Fourier" coefficients associated with the projection of the ratio of the two weights, $w(x)/\hat{w}(x)$ onto the $2N$-dimensional preconditioning orthogonal polynomial subspace, $\mathscr{P}_{\hat{w}}$. The smoother the function $w(x)/\hat{w}(x)$, the more rapid the "modified moment" decays with n. This suggests a way to choose the polynomial subspace $\mathscr{P}_{\hat{w}}$.

When both weights $w(x)$ and $\hat{w}(x)$ are even functions of x on the interval $(-1, 1)$, then both the subspaces $\mathscr{P}_w$ and $\mathscr{P}_{\hat{w}}$ have separable subspaces of even and odd functions. Under these circumstances, we can simplify the "modified moments" algorithm to effect a direct mapping between subspaces of equal parity. This means that there exists a permutation matrix Ω such that

$$T'=\Omega T\Omega^T$$

is a 2×2 block diagonal matrix whose diagonal blocks are lower triangular. The diagonal block matrix represents the transformation between subspaces of equal parity.

The "modified moments" method is reduced to computing the unknowns, β_n, $n=1, 2, ..., N-1$, from the even order "modified moments" $\sigma_{0,2n}$, $n=0, 1, ..., N-1$, since the odd order "modified moments" $\sigma_{0,2n+1}$, $n=0, 1, ...,$ vanish identically. A more compact algorithm can now be derived by taking advantage of the separation of the polynomial subspaces, $\mathscr{P}_w$ and $\mathscr{P}_{\hat{w}}$, into their respective subspaces of even and odd functions.

From (9), we have the three-term recurrence for the even ordered polynomials, $Q_n(x)\equiv \Pi_{2n}(x)$ and $\hat{Q}_n(x)\equiv \hat{P}_{2n}(x)$,

$$Q_{n+1}(x)=(x^2-\tilde{\alpha}_n)\, Q_n(x)-\tilde{\beta}_n Q_{n-1}(x) \tag{29}$$

and

$$\hat{Q}_{n+1}(x)=(\hat{a}_n x^2+\hat{b}_n)\,\hat{Q}_n(x)-\hat{c}_n\hat{Q}_{n-1}(x), \tag{30}$$

where

$$\tilde{\alpha}_n=\beta_{2n}+\beta_{2n+1}, \tag{31}$$

$$\tilde{\beta}_n=\beta_{2n}\beta_{2n-1}, \tag{32}$$

$$\hat{a}_n=a_{2n}a_{2n+1}, \tag{33}$$

$$\hat{b}_n=-c_{2n+1}-c_{2n}a_{2n+1}/a_{2n-1}, \tag{34}$$

and

$$\hat{c}_n=c_{2n}c_{2n-1}a_{2n+1}/a_{2n-1}. \tag{35}$$

The "mixed moments" of the even parity subspaces are

$$\mu_{n,m}\equiv\sigma_{2n,2m}=\int_{-1}^{1}\Pi_{2n}(x)\,\hat{P}_{2m}(x)\, w(x)\, dx = \int_{-1}^{1} Q_n(x)\,\hat{Q}_m(x)\, w(x)\, dx.$$

The "modified moments" algorithm is obtained from (23)–(27) by replacing

$$a_n, b_n, c_n, \alpha_n, \beta_n, \text{ and } \sigma_{n,m}$$

respectively by their corresponding coefficients

$$\hat{a}_n, \hat{b}_n, \hat{c}_n, \tilde{\alpha}_n, \tilde{\beta}_n, \text{ and } \mu_{n,m}.$$

In the above calculation, we have make use of the relation

$$\frac{\gamma_{2n}}{\gamma_{2(n-1)}}=\tilde{\beta}_{2n}.$$

Having computed $\tilde{\alpha}_n$ and $\tilde{\beta}_n$ from the modified moment

algorithm, we may use (31) and (32) to calculate β_{2n} and β_{2n+1} if so desired. Here, we have

$$\beta_1 = \tilde{\alpha}_0,$$
$$\beta_{2n} = \tilde{\beta}_n / \beta_{2n-1},$$

and

$$\beta_{2n+1} = \tilde{\alpha}_n - \beta_{2n}, \qquad n = 1, \ldots, N-1.$$

III.2. *An Application of the Modified Moment Method*

In this subsection, we apply the "modified moments" or the modified Chebyshev's method to the generation of orthogonal polynomials on the finite interval $(-1, 1)$ with respect to the Hermite or Gaussian weight $\exp(-b^2x^2)$. There are three steps in the "modified moments" method: (1) choose a known preconditioning subspace of polynomials that is "close" to the desired orthogonal polynomials; (2) calculate their modified moments with respect to the Gaussian weight; and (3) compute the coefficients α_n and β_n of the three-term recurrence (1) using (23)–(24) and (26)–(27).

We can choose the preconditioning subspace among the known classical orthogonal polynomials subspaces such as, Jacobi, ultraspherical or Gegenbauer, Chebyshev, and Legendre; otherwise the preconditioning orthogonal polynomials need to be generated. Under these circumstances, we are in no better position than we were in the beginning of the discussion. The above known orthogonal polynomial sub-spaces may not provide an adequate preconditioning for all b but they are, nevertheless, accessible and can be easily calculated. For expediency, we choose as the preconditioning polynomial sub-space the set of ultraspherical or Gegenbauer polynomials, C_n^λ. The Gegenbauer polynomials are orthogonal to the weight $(1-x^2)^{\lambda-1/2}$ which depends on a parameter λ. Furthermore, the Legendre polynomials, P_n, and the Chebyshev polynomials, T_n and U_n, are special cases of the Gegenbauer polynomials (Magnus, Oberhettinger, and Soni [13]):

$$C_n^{1/2}(x) = P_n(x),$$
$$C_n^0(x) = \frac{2}{n} T_n(x),$$
$$C_n^1(x) = U_n(x).$$

This permits us to observe the sensitivity of the preconditioning process by merely changing a parameter.

We begin with the computation of the even order modified moments, $\mu_{0,n}$, $n = 0, 1, \ldots, N-1$, associated with the projection of $\exp(-b^2x^2)/(1-x^2)^{\lambda-1/2}$ onto the preconditioning subspace consisting of a set of Gegenbauer polynomials, $C_{2n}^\lambda(x)$:

$$\mu_{0,n} = \int_{-1}^{1} \exp(-b^2x^2)\, C_{2n}^\lambda(x)\, dx. \tag{36}$$

Let

$$v_n^\lambda(b) = \mu_{0,n}(b, \lambda), \qquad n = 0, 1, 2, \ldots, N-1, \tag{37}$$

where N denotes the highest order. An inhomogeneous, linear, three-term recurrence relation can be derived for $v_n^\lambda(b)$,

$$\frac{n+1}{2n+1+\lambda} v_{n+1}^\lambda(b) - \left[\frac{2n+\lambda}{b^2} + \frac{(2n+\lambda)(1-\lambda)}{(2n+\lambda-1)(2n+\lambda+1)}\right] \times v_n^\lambda(b) - \frac{n-1+\lambda}{2n-1+\lambda} v_{n-1}^\lambda(b) = (2\lambda-1)\, C_{2n-1}^\lambda(1) \frac{2n+\lambda}{n(2n+1)} \frac{\exp(-b^2)}{b^2}, \tag{38}$$

where $C_m^\lambda(1) = (2\lambda)_m / m!$ (see Magnus, Oberhettinger, and Soni [13] for the definition of $(\lambda)_m$). The derivation of (38) uses the differentiation formula

$$\frac{d}{dx}[C_{n+1}^\lambda - C_{n-1}^\lambda] = 2(n+\lambda)\, C_n^\lambda$$

and the recurrence relation

$$(n+1)\, C_{n+1}^\lambda = 2(n+\lambda)\, xC_n^\lambda - (n+2\lambda-1)\, C_{n-1}^\lambda, \qquad n = 1, 2, \ldots,$$

for the Gegenbauer polynomials.

It can be readily seen from (38) that when $\lambda = \frac{1}{2}$ or on the set of Legendre polynomials that the inhomogeneous term vanishes identically. For $n \gg 1$ and $n/(b/2)^2 > 1$, then (38) reduces asymptotically to

$$v_{n+1}^\lambda(b) - \frac{2n}{(b^2/2)} v_n^\lambda(b) - v_{n-1}^\lambda(b) = 2(2\lambda-1) \frac{C_{2n-1}^\lambda(1)}{n} \frac{\exp(-b^2)}{b^2}. \tag{39}$$

The homogeneous solutions of (39) are the Bessel functions

$$(-1)^n I_n(b^2/2) \qquad \text{and} \qquad K_n(b^2/2).$$

It is known from Abramowitz and Stegun [14] that $I_n(z)$ decays with n for a fixed z and, therefore, is the subdominant solution of the homogeneous part of (38). Similarly, $K_n(z)$ grows with n for a fixed z and, therefore, is the dominant solution. We seek, however, the subdominant solution and, thus, this renders a forward marching method unacceptable in the presence of the dominant solution which can be excited by numerical perturbations. This suggests that we use Olver's method [15] for computing the solution of (38) given the initial conditions $v_0^\lambda(b)$ and $v_1^\lambda(b)$.

The essence of Olver's method is to convert an initial value problem into a two point boundary value problem by disregarding the value of $v_1^\lambda(b)$ and imposing, instead, a homogeneous boundary condition $v_M^\lambda(b)=0$ for some arbitrary M. The value of M is determined by ensuring that the solution will be essentially unaffected by prolongating the imposition of the artificial homogeneous boundary condition by one more order. In effect, the decay rate of the subdominant solution is estimated. Rather than faithfully follow the details of Olver's method, we develop, in what follows, a discrete invariant imbedding variant of Olver's algorithm. It is akin to that proposed by Van der Cruyssen [16] who employs a LU factorization of the resulting tridiagonal coefficient matrix.

The system of linear three-term recurrence relations, (38),

$$c_n v_{n+1}^\lambda(b) - b_n v_n^\lambda(b) + a_n v_{n-1}^\lambda(b) = d_n,$$
$$n = 1, 2, ..., M,$$

with boundary conditions,

$$v_0^\lambda(b) = A \qquad \text{and} \qquad v_{M+1}^\lambda(b) = 0,$$

when written in matrix notation, has a tridiagonal coefficient matrix $\hat{A}$. The vector of unknowns, $v_n^\lambda(b)$, $n = 1, 2, ..., M$, is ordered naturally.

Applying the discrete invariant embedding algorithm to its solution, we obtain the decomposition

$$v_i^\lambda(b) = R_i v_{i+1}^\lambda(b) + S_i, \qquad i = 1, 2, ..., M; \tag{40}$$

$$R_i = \frac{c_i}{b_i - a_i R_{i-1}},$$
$$i = 2, 3, ..., M, \text{ with } R_1 = c_1/b_1; \tag{41}$$

and

$$S_i = \frac{R_i}{c_i}(a_i s_{i-1} - d_i), \qquad i = 2, 3, ..., M,$$
$$\text{with } S_1 = -\frac{d_1 - a_1 A}{b_1}; \tag{42}$$

Equations (41)–(42) constitute the forward sweep and (40) is the backward sweep.

Next we turn to the question of choosing M: if M is chosen too large, wasteful computation results; and if M is too small, inaccuracy is obtained. Olver suggests the following method for determining the optimal value of M. The key is to observe the effect on the solution $v_L^\lambda(b)$ for some $L < M$ as a result of prolongating the imposition of the artificial homogeneous boundary condition by one more order, i.e., setting $v_{M+2}^\lambda(b) = 0$ instead of $v_{M+1}^\lambda(b) = 0$. This also yields an estimate of the rate of decay of the inverse of the coefficient matrix, $\hat{A}$. The result of the calculation is given by

$$|v_L^\lambda(b, M+2) - v_L^\lambda(b, M+1)| = \left| S_{M+1} \prod_{k=L}^{M} R_k \right|. \tag{43}$$

Suppose we wish to compute $v_L^\lambda(b)$ to D decimal places for given values of L and D; then setting

$$\left| S_{M+1} \prod_{k=L}^{M} R_k \right| < \frac{10^{-D}}{2}$$

would ensure that $v_L^\lambda(b, M+1)$ and $v_L^\lambda(b, M+2)$ obtained by solving (40), in the backward direction, agree to D decimal places.

The computational algorithm consists of fixing the values of D and L and of performing the forward sweep (41)–(42). During the performance of the forward sweep with $n > L$, additionally we compute

$$\Delta_n = S_n \prod_{k=L}^{n-1} R_k, \qquad n = L+1, ...$$

If $|\Delta_n| < 10^{-D}/2$, we set $M = n$ and perform the backward sweep (40) with $i = M-1, M-2, ..., 1$.

IV. A DOMAIN DECOMPOSITION METHOD

Domain decomposition methods are becoming increasingly popular for solving partial differential equations. The reason lies in their adaptivity to problems with many and differing local structures. The essence of this class of methods is the patching of differing local sub-domain solutions to form a global solution. Another laudable feature of the domain decomposition methods is that in each sub-domain we may use an entirely different solution technique. This allows for increased efficiency and accuracy. In the present setting, we perform domain decomposition on the two-dimensional parameter space of b, a parameter associated with the Gaussian weight, and of n, the order of the orthogonal polynomial. These parameters appear

naturally in our formulation. Clearly, this method can be fruitfully applied to generating orthogonal polynomials with a positive weight on a finite interval.

Heuristically, as we examine the weight function,

$$\exp\{-(bx)^2\},$$

we see that for $b<1$ and $|x|<1$ the leading term of the Taylor expansion is 1 which corresponds to an uniform weight. Associated with the uniform weight is the set of Legendre polynomials $P_n(x)$. Thus for $b<1$ and $\forall n$, the orthogonal polynomials of interest behave like the Legendre polynomials.

Since an orthogonal polynomial of order n has n zeroes within the interval of definition, a comparison of the oscillation scale with that of the scale of variation of the Gaussian weight suggests, once again, that $\Pi_n(x)$ resembles the Legendre polynomials for $n \gg b^2$.

For $b \gg 1$, we rescale x so that the Gaussian weight becomes

$$\exp\{-y^2\}$$

and the limits of integration go to $\pm b$. Here, it is obvious that the polynomials $\Pi_n(x)$ are asymptotically closed in behavior to that of the Hermite polynomials $H_n(y)$ or, equivalently, $H_n(bx)$.

Based on a comparison of their scales with that of the weight, we may expect, for any finite b, the following behavior of the polynomials: (1) for $0<n<N^*$, $\Pi_n(x) \sim H_n(bx)$, where N^* is that highest possible order of Hermite polynomial whose oscillations are entirely confined within the basic interval $(-1, 1)$, and (2) for $n \gg b^2$, $\Pi_n(x) \sim P_n(x)$. In between, we have a transitional region.

These arguments will be made more definite in what follows. Using the definition of the weighted norm (3) and the recurrence relation (7), we can rewrite the weighted norm γ_n as

$$\gamma_n = \int_{-1}^{1} x\Pi_{n-1}(x)\,\Pi_n(x)\,w(x)\,dx.$$

Integrating the right-hand side by parts and using both the definition of the monic polynomial and the reflection symmetry of $w(x)$ about the origin, we derive the following two-term recurrence relation for γ_n,

$$\gamma_n = \frac{n}{2b^2}\gamma_{n-1} - \frac{\exp(-b^2)}{b^2}\Pi_{n-1}(1)\,\Pi_n(1), \tag{44}$$

where we have used $\Pi_n(-1) = (-1)^n\,\Pi_n(1)$. From (44), we see immediately that forward marching with increasing n is stable if $n/2b^2 < 1$ and, conversely, backward recurrence with decreasing n is stable for $n/2b^2 > 1$.

An equivalent forward marching *Stieltjes procedure* can be established for calculating $\Pi_n(1)$, γ_n, and β_n by using (44), the definition of β_n, and (7) restricted to $x=1$, i.e.,

$$\Pi_{n+1}(1) = \Pi_n(1) - \beta_n\Pi_{n-1}(1). \tag{45}$$

The starting values are $\Pi_{-1}(1)=0$ and $\Pi_0(1)=1$. Unfortunately, this process, i.e., the *Stieltjes procedure* is numerically unstable for all values of b and n.

For problems with exponential weights or Freud weights, the "natural" orthonormal polynomials are more useful (see Nevai [17]). They are obtainable from the monic polynomials by a simple rescaling. The three-term recurrence in this case becomes

$$a_{n+1}q_{n+1}(x) = xq_n(x) - a_nq_{n-1}(x). \tag{46}$$

Here,

$$\int_{-1}^{1} q_n^2(x)\,w(x)\,dx = 1,$$

$$\Pi_n(x) = \sqrt{\gamma_n}\,q_n(x),$$

and

$$a_n = \sqrt{\beta_n}.$$

Moreover,

$$\gamma_n = \frac{n}{2b^2}\gamma_{n-1} - \frac{\exp(-b^2)}{b^2}\Pi_{n-1}(1)\,\Pi_n(1),$$

and $$a_n = \delta_{n-1}/\delta_n.$$

Following Nevai, a pair of nonlinear recurrence is derived for a_n and $q_n(1)$:

$$\frac{n}{a_n} = 2q_{n-1}(1)\,q_n(1)\exp(-b^2) + 2b^2a_n \tag{47}$$

and

$$2n+1 = 2q_n^2(1)\exp(-b^2) + 2b^2(a_n^2 + a_{n+1}^2). \tag{48}$$

In deriving (47)–(48), we have used $q_n(-1) = (-1)^n\,q_n(1)$. Moreover, it should be noted that (47) is directly obtainable from (44) using the aforementioned scaling. When both (47) evaluated at n and at $n+1$ are substituted into (48), we obtain

$$a_{n+1}q_{n+1}(1) = q_n(1) - a_nq_{n-1}(1). \tag{49}$$

Therefore, the set of Eq. (47)–(49) is not independent. We may choose any pair.

The utility of (47)–(49) for analysis is immediately evident as we can simply readoff the results, as $b \to 0$,

$$2n+1 = 2q_n^2(1) \tag{50}$$

and

$$\frac{n}{a_n} = 2q_{n-1}(1)\, q_n(1). \tag{51}$$

They describe the set of orthonormal Legendre polynomials. Similarly, as $b \to \infty$, we have

$$a_n^2 = n/2b^2 \tag{52}$$

and

$$\sqrt{(n+1)/2}\, q_{n+1}(1) = b\, q_n(1) - \sqrt{n/2}\, q_{n-1}(1). \tag{53}$$

The solution of (53) is the orthonormal Hermite polynomials evaluated at $x = b$, namely $h_n(b)$. Since an orthogonal polynomial of order n executes n oscillations in the interval of definition and $\delta_n > 0$ for all n, we must have $q_n(1) > 0$ (Nevai [18]). These same arguments, namely the oscillations of the orthogonal polynomials and the positivity of $q_n(1)$, can be used to find an upper bound on n called N^* above, of which the orthogonal polynomials of interest deviate significantly from the set of Hermite polynomials.

The following bounds on a_n uniform for b and n can be established from the positivity of $q_n(1)$ for $\forall n$ and from an analysis of (47)–(48):

$$\sqrt{\frac{n}{2b^2}} > a_n > \left(\frac{n}{\sqrt{4n^2-1}}\right) \times \left[\frac{2}{1+\sqrt{1+8b^2n/(4n^2-1)}}\right]. \tag{54}$$

To refine the estimates of (50)–(52), we introduce the change of variables,

$$a_n = \frac{n}{\sqrt{(2n-1)(2n+1)}}\,\alpha_n \tag{55}$$

and

$$q_n(1) = \sqrt{(2n+1)/2}\, \exp\left(-\frac{b^2}{2}\right) r_n. \tag{56}$$

The transformed equations are

$$\frac{1}{\alpha_n} = r_{n-1} r_n + \frac{2b^2}{2n+1}\left(\frac{n}{2n-1}\right)\alpha_n \tag{57}$$

and

$$r_n^2 = 1 - \frac{2b^2}{2n+1}\left[\frac{(n+1)^2}{(2n+1)(2n+3)}\,\alpha_{n+1}^2 + \frac{n^2}{4n^2-1}\,\alpha_n^2\right]. \tag{58}$$

It can be seen from (57)–(58) that

$$\frac{1}{2} < \frac{n}{2n-1} \leqslant 1,$$

$$\frac{1}{4} < \frac{(n+1)^2}{(2n+1)(2n+3)} \leqslant \frac{4}{15},$$

$$\frac{1}{4} < \frac{n^2}{4n^2-1} \leqslant \frac{1}{3}$$

and, therefore, for

$$\frac{2b^2}{2n+1} \ll 1$$

$$r_n = 1 - \mathcal{O}\left(\frac{2b^2}{2n+1}\right),$$

$$\alpha_n = 1 + \mathcal{O}\left(\frac{2b^2}{2n+1}\right).$$

This estimate together with the bounds established earlier proves the assertion that for $b \ll 1$ and $n \geqslant 1$ such that $2b^2/(2n+1) \ll 1$ the orthogonal polynomials associated with the Gaussian weight is asymptotically similar to that of the Legendre polynomials. In fact, the orthonormal Rys polynomials, $q_n(x)$, are slight perturbations of the Legendre polynomials, $P_n(x)$. This also suggests that the nonlinear Gauss–Seidel iterative method given by

$$\alpha_n^{(k)} = 2/\{r_{n-1}^{(k-1)} r_n^{(k-1)} + \sqrt{[r_{n-1}^{(k-1)} r_n^{(k-1)}]^2 + 4(2b^2/(2n+1))(n/(2n-1))}\},$$

$$r_n^{(k)^2} = 1 - \frac{2b^2}{2n+1}\left[\frac{(n+1)^2}{(2n+1)(2n+3)}\,(\alpha_{n+1}^{(k)})^2 + \frac{n^2}{4n^2-1}\,(\alpha_n^{(k)})^2\right],$$

and

$$r_n^{(0)} = 1$$

is convergent for $2b^2/(2n+1) \ll 1$. We remark that for any fixed b the nonlinear Gauss–Seidel iterative scheme generates a perturbation solution to (57)–(58) with respect to the parameter $2b^2/(2n+1)$. The sequence of partial sums is weakly dependent on $1/n$. Thus, the nonlinear Gauss–Seidel iterative solution is more rapidly convergent

with increasing n. This observation will be utilized later in our algorithm.

Similar results can be obtained for $b \gg 1$. Here, (52)–(53) suggest the introduction of d_n and s_n,

$$a_n = \frac{d_n \sqrt{n/2}}{b} \tag{59}$$

and

$$q_n(1) = \sqrt{b/\mathrm{erf}\{b\}}\, s_n, \tag{60}$$

where $\mathrm{erf}\{x\}$ is the error function. Substituting these changes of variables into (46) and (47), we obtain

$$1 = 2\varepsilon \frac{s_{n-1} s_n}{\sqrt{2n}} d_n + d_n^2 \tag{61}$$

and

$$\sqrt{(n+1)/2}\, d_{n+1} s_{n+1} = b\, s_n - \sqrt{n/2}\, d_n s_{n-1}, \tag{62}$$

where

$$\varepsilon = \exp(-b^2)/\mathrm{erf}\{b\}.$$

In contrast to the previous situation, it is necessary to solve the three-term recurrence for s_n. The starting values for the forward marching procedure are

$$s_{-1} = 0 \qquad \text{and} \qquad s_0 = 1/\pi^{1/4}. \tag{63}$$

We note that the normalization appearing in (60) is introduced for convenience so that s_0 is numerically near one.

Another nonlinear Gauss–Seidel iterative method can be constructed for the solution of (61)–(63):

$$\sqrt{(n+1)/2}\, d_{n+1}^{(k-1)} s_{n+1}^{(k)} = b\, s_n^{(k)} - \sqrt{n/2}\, d_n^{(k-1)} s_{n-1}^{(k)} \tag{64}$$

with initial values

$$s_{-1}^{(k)} = 0 \qquad \text{and} \qquad s_0^{(k)} = \frac{1}{\pi^{1/4}} \tag{65}$$

and

$$d_n^{(k)} = 1 \Big/ \left(\sqrt{1 + \frac{\varepsilon^2}{2n} [s_{n-1}^{(k)} s_n^{(k)}]^2} + \frac{\varepsilon}{\sqrt{2n}} s_{n-1}^{(k)} s_n^{(k)} \right). \tag{66}$$

Here, we set $d_n^{(0)} = 1$.

From a previous discussion, the solution $s_n^{(0)}$ is just $h_n(b)$, the orthonormal Hermite polynomials of order n evaluated at b. Since $h_n(bx)$ are oscillatory in $(-1, 1)$ for $b^2 > 2n+1$, we must have $N^* = \lfloor (b^2-1)/2 \rfloor$, where $\lfloor\ \rfloor$ denotes the integer value. To see this, we note that the usual Hermite polynomial $H_n(y)$, $-\infty < y < \infty$ satisfies a second-order Sturm–Liouville equation of the form

$$u''(y) + (2n + 1 - y^2)\, u(y) = 0,$$

where

$$u(y) = \exp(-y^2/2)\, H_n(y).$$

Thus, $H_n(bx)$ is non-oscillatory in $(-1, 1)$ if $b^2 > 2n+1$. Alternatively, (53) written in matrix form will have a tridiagonal, symmetric coefficient matrix, called a Jacobi matrix. The coefficient matrix is diagonally dominant if $b^2 > 2n+1$. It follows that the initial value problem has a discrete maximum principle and, therefore, there is a dominant solution and a sub-domainant solution. The present discussion, together with the considerations following (44), suggests that forward marching is a viable solution technique for $b > 1$ and $1 \leqslant n \leqslant N^*$. This is the topic of discussion of the next section.

V. THE ALGORITHM

The algorithm is separated into three parts corresponding to changes in the local behaviors of the solution. We perform forward marching for $1 \leqslant n \leqslant \hat{N}$, where $\hat{N}$ is to be determined, then a two-point boundary-value problem solver is used to continue for $\hat{N}+1 \leqslant n \leqslant M$, where M can be the order of the highest polynomial desired. The boundary condition at $M+1$ is obtained by applying a nonlinear Gauss–Jacobi iterative scheme to the intermediate equation to be discussed in what follows. The method of solution for the two-point boundary-value problem is akin to Olver's method discussed in Section III.2.

The *forward marching* method makes use of (61) and (62) with initial conditions (63). They are rewritten here for convenience,

$$d_n = \sqrt{1 - 2(b\varepsilon/n)\, s_{n-1} [s_{n-1} - \sqrt{(n-1)/2}\, d_{n-1} s_{n-2}]}, \tag{67}$$

and

$$s_n = \frac{b\sqrt{2/n}}{d_n} [s_{n-1} - \sqrt{(n-1)/2}\, d_{n-1} s_{n-2}]. \tag{68}$$

Using double precision arithmetic to evaluate expressions (67)–(68), we can extend the range of the computation beyond $N^* = \lfloor (b^2-1)/2 \rfloor$, an estimate obtained from perturbation theory. The new limit, the switching point $\hat{N}$ for a given value of b, is derived from computational experience.

This is done by comparing solutions of forward marching and of the domain decomposition with the switching point set at N^*. The final switching points are chosen so that the recurrence coefficients have full single machine precision (14 significant figures on Cray machines).

To calculate the recurrence coefficients a_n beyond the switching point $\hat{N}$, a two-point boundary-value problem must be formulated to circumvent numerical error growth of the forward marching scheme. An equation involving only the coefficients, a_n can be derived by squaring (47) and by replacing $q_n^2(1)$ and $q_{n-1}^2(1)$ thus appear with (48). The final result is

$$\begin{aligned} n^2 - 4nb^2e_n + 4b^4e_n^2 &= e_n[2n+1-2b^2(e_n+e_{n+1}) \\ &\quad \times [2n-1-2b^2(e_n+e_{n-1})], \end{aligned} \tag{69}$$

where

$$e_n = a_n^2.$$

Further reduction can be made by introducing the following change of variable,

$$e_n = \tfrac{1}{4}(1+\psi_n/F_n) \tag{70}$$

to yield

$$\begin{aligned} s_n = 0 = &-\frac{b^2G_n}{2F_{n-1}}\psi_{n-1} + \psi_n - \frac{b^2G_{n-1}}{2F_{n+1}}\psi_{n+1} \\ &- \bar{s}_n(\psi_{n-1}, \psi_n, \psi_{n+1}), \end{aligned} \tag{71}$$

where

$$\begin{aligned} &\bar{s}_n(\psi_{n-1}, \psi_n, \psi_{n+1}) \\ &= 1 + \frac{b^2}{F_n}\left[\frac{2n}{F_n}\psi_n + \left(\frac{G_n}{F_{n-1}}\psi_{n-1} + \frac{G_{n-1}}{F_{n+1}}\psi_{n+1}\right)\right]\psi_n \\ &\quad - \left(\frac{b^2}{2F_n}\right)^2\left[1+\frac{\psi_n}{F_n}\right]\left[\psi_n + \frac{F_n}{F_{n-1}}\psi_{n-1}\right] \\ &\quad \times \left[\psi_n + \frac{F_n}{F_{n+1}}\psi_{n+1}\right], \end{aligned} \tag{72}$$

$$F_n = 2n(2n-b^2)-1,$$

$$\text{and} \qquad G_n = 2n - b^2 + 1. \tag{73}$$

We note that the leading term, $\frac{1}{4}$ is suggested by Nevai's asymptotic analysis [17] and that F_n is singular for

$$n = \tilde{N} = \frac{b^2 + \sqrt{b^4+1}}{4}.$$

Therefore, this formulation is valid for $n > \tilde{N}$. Moreover, when $b = 0$ we have

$$\psi_n \equiv 1 \qquad \text{and} \qquad F_n = 4n^2 - 1,$$

leading to $e_n = n^2/(4n^2-1)$, which are the recurrence coefficients for Legendre polynomials. This suggests that with the present setting, (70)–(73), the calculated solution is uniform in b. Therefore, it is not necessary for computational purposes to explicitly identify the Legendre polynomials as discussed previously.

To simplify the notation, let

$$\begin{aligned} \Psi &= (\psi_{\hat{N}+1}, \psi_{\hat{N}+2}, \ldots, \psi_M)^T, \\ S &= (s_{\hat{N}+1}, s_{\hat{N}+2}, \ldots, s_M)^T, \\ \bar{S} &= (\bar{s}_{\hat{N}+1}, \bar{s}_{\hat{N}+2}, \ldots, \bar{s}_M)^T \end{aligned}$$

be $\mu = M - \hat{N}$ component vectors and A, a $\mu \times \mu$, tridiagonal matrix

$$\begin{bmatrix} \ddots & \ddots & & & & & \\ \ddots & \ddots & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & -\dfrac{b^2G_n}{2F_{n-1}} & 1 & -\dfrac{b^2G_{n-1}}{2F_{n+1}} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \ddots & \ddots \end{bmatrix}, \tag{74}$$

then (71) may be rewritten in matrix form as

$$0 = S(\Psi) = A\Psi - \bar{S}(\Psi).$$

We remark that (69)–(70) evaluated at $n = \hat{N}$ and M involve respectively the solutions at both endpoints, $\psi_{\hat{N}}$ and ψ_{M+1}. Thus, (71) is a two-point boundary-value problem. These values must be provided for. The completion of the forward marching scheme (67)–(68) yields one boundary value, $\psi_{\hat{N}}$. For the other boundary value ψ_{M+1}, it will be computed by using an iterative procedure different from that employed in solving (71). As a result, there are three distinct subdomains in which we employed different solution techniques and different formulations of the fundamental equations. This represents the differing inherent local behaviors of the Rys polynomial. A domain decomposition method can accommodate conveniently and naturally these changes in the formulation of the computational problem as well in its solution technique.

To continue with the algorithm development, we note

that the off-diagonal elements of the tridiagonal matrix A have the bounds

$$\frac{b^2G_n}{2F_{n-1}} < \frac{b^2G_{\hat{N}+2}}{2F_{\hat{N}+1}} < \frac{1}{2}$$

$$\frac{b^2G_{n-1}}{2F_{n+1}} < \frac{b^2(G_n-2)}{2F_{n-1}} < \frac{b^2G_n}{2F_{n-1}}$$

for

$$n > \hat{N} \geqslant \frac{b^2+\sqrt{3b^2+1}}{2} - 1. \tag{75}$$

It follows that A is irreducibly diagonally dominant with positive diagonal and negative off-diagonal entries; hence, its inverse is non-negative and non-singular. In fact, A is a M-matrix. The spectral radius of $I-A$ is less than one, i.e., $\rho(I-A)<1$ [19].

Moreover, if the inequality (75) holds, then we have

$$|\bar{S}(x)-\bar{S}(y)| < \kappa\,|x-y|, \qquad \kappa<1 \tag{76}$$

for

$$|x| \leqslant 1 \qquad \text{and} \qquad |y| \leqslant 1.$$

This ensures that the vector-valued function, $\bar{S}$: $R^\mu \to R^\mu$ is locally contractive. These considerations lead immediately to the convergence of the following modified Newton's method:

$$A[\Psi^{(k)}-\Psi^{(k-1)}] = -S(\Psi^{(k-1)}) \qquad \text{with} \quad \Psi^{(0)} = I$$

which can be written simply as

$$A\Psi^{(k)} = \bar{S}(\Psi^{(k-1)}). \tag{77}$$

This is used in our subsequent computations.

An examination of (72)–(73) shows that the Lifshitz constant κ associated (76) is

$$O(b^2/n).$$

Therefore, the nonlinear Gauss–Jacobi iterative method,

$$\Psi^{(k)} = (L+U)\,\Psi^{(k-1)} + \bar{S}(\Psi^{(k-1)}),$$
$$\text{with} \quad \Psi^{(0)} = I; \tag{78}$$

where L and U are respectively strictly lower and upper, $\mu\times\mu$, triangular matrix obtained from a regular splitting of A;

$$A = I-(L+U),$$

is convergent for $n > N_c = (b^2/c)$, where $c<1$ is some constant independent of n.

One possibility of calculating ψ_{M+1}, the boundary value at $n=M+1$, is to choose $M > N_c$ and apply the nonlinear Gauss–Jacobi iterative method (78). In this region of the parameter space, the nonlinear Gauss–Jacobi iterative method generates a finite perturbation series expansion of ψ_n in terms of b^2/n. To achieve $O((b^2/n)^L)$ accuracy about $n=M+1$, $2L+1$ values of $\psi_n^{(0)}$ are needed. The Gauss–Jacobi iterative method starting with $\psi_n^{(0)}=1$, $n=M+1-L, ..., M+1, ..., M+1+L$ pyramids to a single final value of ψ_{M+1} with the desired accuracy. If the asymptotic solution for $n \gg b^2$ is sought, this is the natural method for generating the higher order terms.

Another possibility is to applied Olver's method discussed in Section III to determine the effect of a perturbation in the boundary value on the solution in the interior. The boundary truncation estimate of Olver will, indeed, provide such a rate of error propagation. A third possibility is to combine these two methods so that the final accuracy is at full machine precision. This is our method in the outer domain. In particular, we set $L=5$ and use Olver's estimate to determine the extension of the boundary. In general, computational experience has shown that four to five additional steps are needed.

To further accelerate the computation, we make full use of an earlier observation that the convergence of the nonlinear Gauss–Seidel iterative method is more rapid with increasing n to terminate the iterative solution of (77). Here, as the iteration proceeds, the solution vector becomes shorter as the higher order components converge faster. This is another example of using all of the available analytic knowledge to increase the efficiency of the numerical method.

It is also interesting to note that (70) suggest that for $n \approx b^2 \gg 1$ there is another orthogonal polynomial subspace to which the Rys polynomials are asymptotically related. This subspace is given by $a_n = \frac{1}{2}$ for some set of n. Since both Chebyshev polynomials of the first and the second kind are identified with $a_n = \frac{1}{2}$, the distinguishing feature is, then, their value at $x=1$. Since Chebyshev polynomials of the second kind increase linearly with n there, they are the polynomials of choice. This exercise tends to show the synergism between computation and analysis, namely, that analysis leads to the formulation of the domain decomposition method which in the term leads to the discovery of the Chebyshev polynomials of the second kind as the transitional polynomial space between the Hermite polynomials for $n < \hat{N}$ and the Legendre polynomials for $n \gg b^2$.

VI. COMPUTATIONAL RESULTS AND DISCUSSION

Both the "modified moments" method of Section III.2 and the domain decomposition method have been programmed in FORTRAN and executed on a CRAY computer. The domain decomposition results are accurate to full machine precision of 14 significant figures. To acertain the accuracy of the solution by the domain decomposition method, we compare them with the solutions obtained by the convergent Gauss–Seidel iterative methods in their respective regions when possible.

The switching point, $\hat{N}$ which separates the *Stieltjes* procedure from the two-point boundary-value method is

$$\hat{N} = \begin{cases} 7 + \text{nint}\{7.8 \times (b - 1.5)\} & \text{for } 1.5 < b \leqslant 3.0 \\ 19 + \text{nint}\{9.6 \times (b - 3.0)\} & \text{for } 3.0 < b \leqslant 8.0 \\ 67 + \text{nint}\{16.5 \times (b - 8.0)\} & \text{for } 8.0 < b \leqslant 10.0 \\ 100 + \text{nint}\{12.2 \times (b - 10.0)\} & \text{for } 10.0 < b \leqslant 15.0 \\ \text{nint}\{b^2/2 + 12.5\sqrt{b}\} & \text{for } 15.0 < b, \end{cases}$$

where nint$\{z\}$ denotes nearest integer value of z. These values are obtained by comparing the solutions of the domain decomposition method with that of the Stieltjes procedure so that the $a_{\hat{N}}$ has full machine precision. It is worth noting that the switching point so obtained is well into the convergent regions of both the modified Newton's method (77) and the nonlinear Gauss–Jacobi iterative method (78) for solving (71).

TABLE I

Recurrence Coefficients

n	$b = 1.0$	4.0	7.0	10.0
1	0.50369048214522	0.17677665040159	0.10101525445522	0.07071067811865
2	0.52487719608986	0.24999901588434	0.14285714285714	0.10000000000000
3	0.51174521082925	0.30617456640840	0.17496355305594	0.12247448713916
4	0.50559349008529	0.35346277600225	0.20203050891044	0.14142135623731
5	0.50321922418396	0.39478178904026	0.22587697572631	0.15811388300842
6	0.50211240053702	0.43093140632470	0.24743582965269	0.17320508075689
7	0.50149940388810	0.46110486434426	0.26726124191236	0.18708286933870
8	0.50112121054601	0.48355962976400	0.28571428571354	0.20000000000000
9	0.50087064887162	0.49727394650698	0.30304576335840	0.21213203435596
10	0.50069585696553	0.50339106385693	0.31943828243850	0.22360679774998
15	0.50029775338437	0.50146513986250	0.39122995082961	0.27386127875258
20	0.50016449692731	0.50044490737323	0.45151534535415	0.31622776601684
25	0.50010417182890	0.50021805570984	0.49478835770008	0.35355339059327
30	0.50007184023604	0.50012962760449	0.50204022152527	0.38729833460953
35	0.50005252113442	0.50008591452018	0.50064374930540	0.41833000170616
40	0.50004006396268	0.50006111386882	0.50026955749509	0.44721022713594
45	0.50003156546726	0.50004569262037	0.50015082117038	0.47407710634009
50	0.50002551002708	0.50003545188179	0.50009680514051	0.49506996605430
60	0.50001765524425	0.50002312143805	0.50004972445119	0.50075227119063
70	0.50001293987793	0.50001626235739	0.50003022787626	0.50016149330894
80	0.50000988918701	0.50001205792713	0.50002030492757	0.50007006104567
90	0.50000780271018	0.50000929592895	0.50001457428051	0.50003920668490
100	0.50000631310588	0.50000738472064	0.50001096760512	0.50002504677066

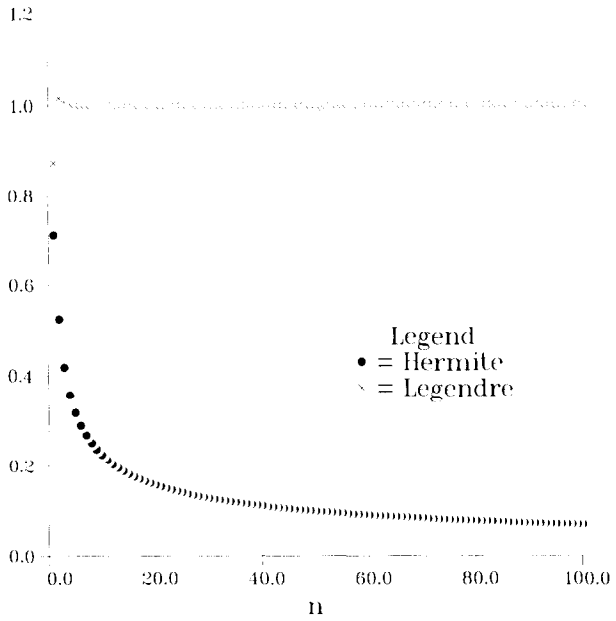


FIG. 1. $b = 1.0$.

The computed recurrence coefficients for $b = 1$, 4, 7, 10 are shown in Table I. For this set of values of b, we have also plotted respectively in Figs. 1–4 the ratios of a_n/a_n^H and a_n/a_n^L, where a_n^H are the recurrence coefficients associated with the Hermite polynomials and a_n^L are the coefficients associated with the Legendre polynomials. These plots allow us to see explicitly the identity of the subspaces and the length of the transition zone. Indeed, the separation of the subspaces is clearly delinated as is the progressive reproportioning of the subspaces with increasing b. This

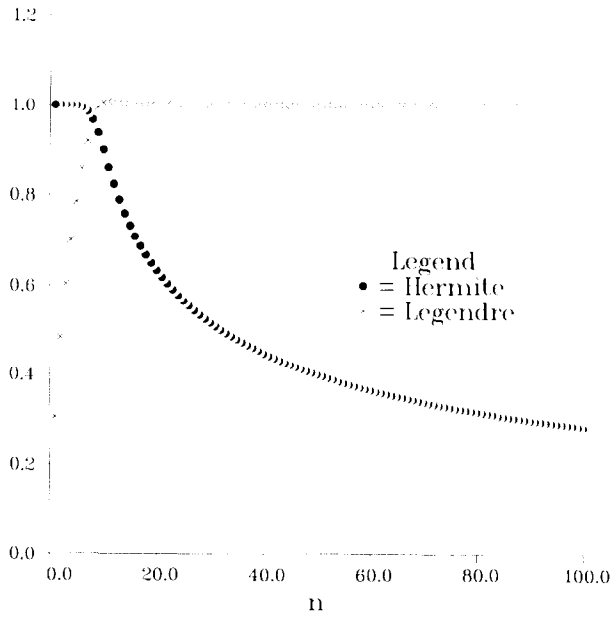


FIG. 2. $b = 4.0$.

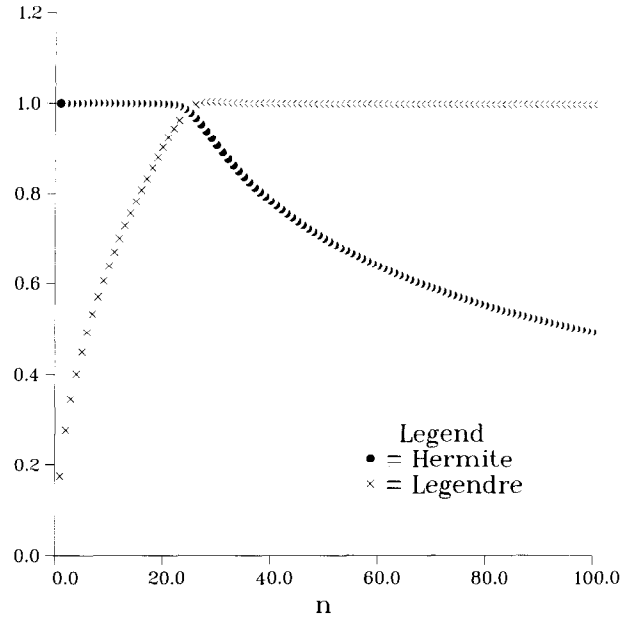
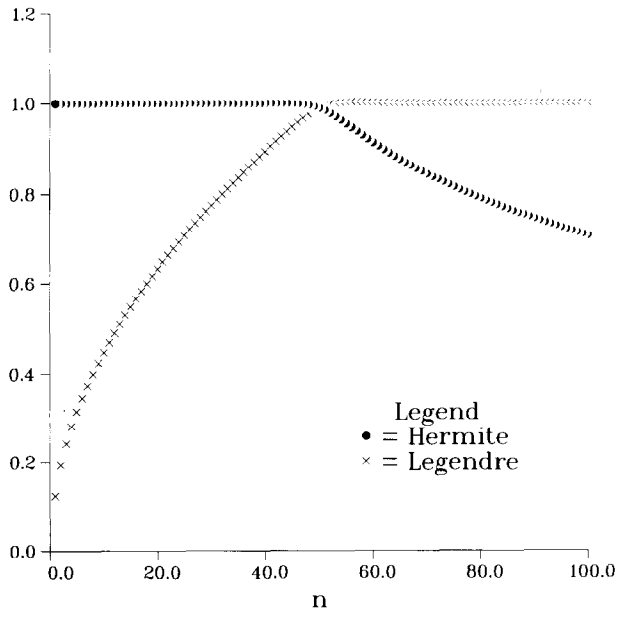


FIG. 3. $b = 7.0$.

tends to suggest that the "modified moments" method using just one preconditioning subspace, say the Legendre polynomials, will eventually fail since the Rys polynomials do change character with b and n as predicted by our analysis.

A series of computations is made with the "modified moments" method developed in Section III.2 using the Gegenbauer polynomials as the preconditioner. The objective of these calculations is to assess the effectiveness of the various preconditioning polynomial subspaces. Recalling that the Legendre polynomials and the Chebyshev polynomials of the first and second kind are special cases of the



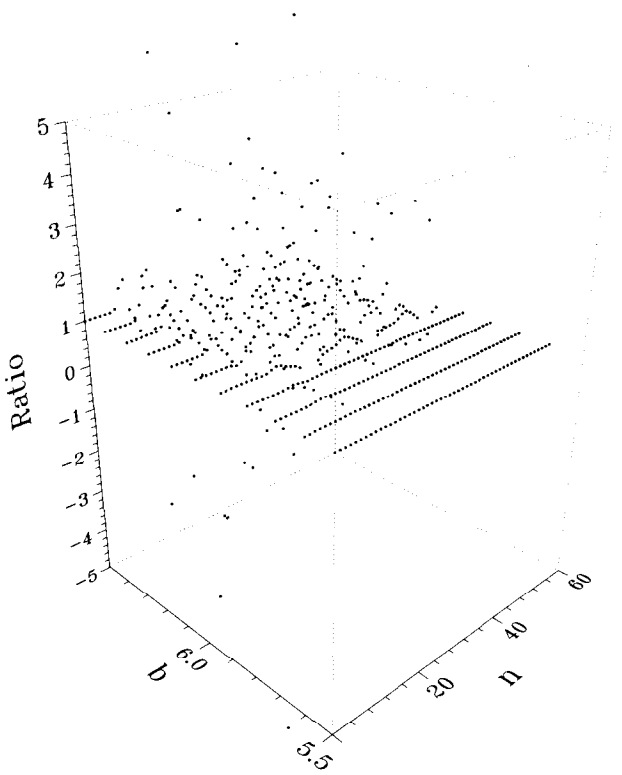


FIG. 5. Chebyshev I.

Gegenbauer polynomials and that the associated weights depend on a parameter, we can examine a host of possible preconditioners by merely changing the parameter, λ. It is found that the "modified moments" method using single precision arithmetic produces full machine accuracy for $b \leqslant 3$. Then, the calculations begin to lose significance. Nine-place accuracy is obtained for $b = 4$. Thereafter loss of significance increases. Figures 5–7 document respectively the degradation of the numerical solution as b increases for the three preconditioning classical orthogonal polynomials; Chebyshev polynomials of the first kind, the Legendre polynomials, and the Chebyshev polynomials of the second kind. Since this paper is concerned mainly with the development of an *effective* method for generating orthogonal polynomials given the Gaussian weight there is no extra effort put into the computation of the "modified moments," v_n^{λ}, nor put into investigating further the cause of the degradation of the solution.

To ensure that the preconditioning polynomial space is "close" to the desired orthogonal polynomials, we propose a continuation method in which the previous solution at some $b = b_1$, i.e., $q_n(x, b_1)$ is used to precondition the computation at some other $b \neq b_1$. The explicit dependence of the orthogonal polynomials on the parameter b is displayed for clarity of exposition. The idea underlying the continua-

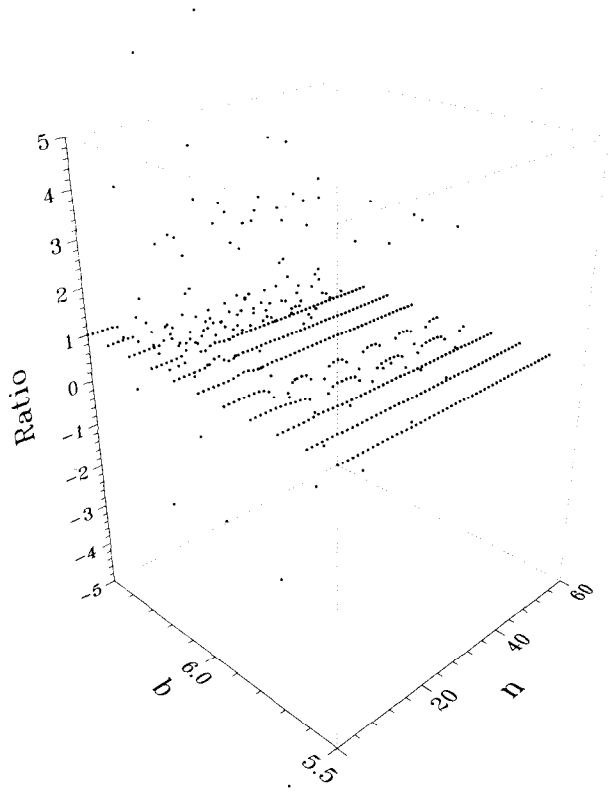


FIG. 6. Legendre

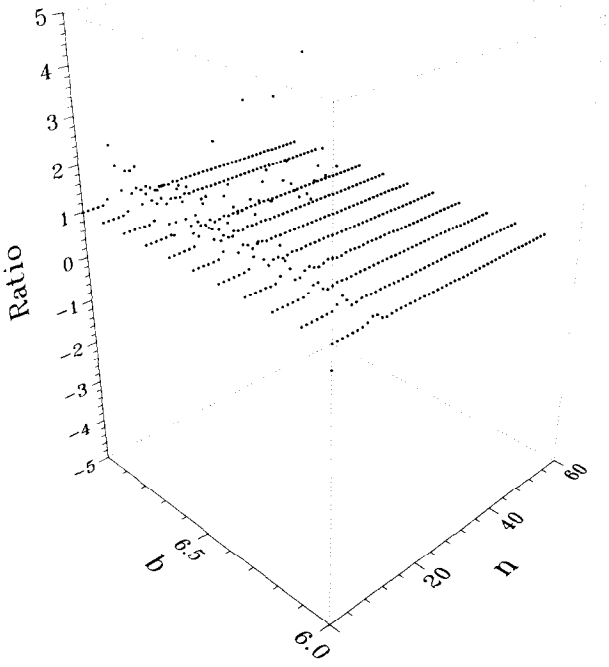


FIG. 7. Chebyshev II.

for the "modified moments" in which the solution at b_1 is the preconditioner. In this case, the "modified moments" are

$$v_n(b, b_1) = \int_{-1}^{1} e^{-b^2x^2} q_n(x, b_1)\, dx$$

$$= \int_{-1}^{1} e^{-(b^2 - b_1^2)x^2} q_n(x, b_1)\, e^{-b_1^2x^2}\, dx,$$

which we can rewrite with the introduction of $c = b^2 - b_1^2$ to yield

$$v_n(c, b_1) = \int_{-1}^{1} e^{-cx^2} q_n(x, b_1)\, e^{-b_1^2x^2}\, dx. \tag{77}$$

Here, $v_n(c, b_1)$ is the "Fourier" coefficient of e^{-cx^2} projected onto the orthogonal polynomial subspace of $\{q_n(x, b_1)\}$. Note that only the even ordered "modified moments" are nonvanishing.

Since the integral is well defined, we may differentiate inside the integral with respect to c to obtain

$$v'_{2m}(c) = -\int_{-1}^{1} e^{-cx^2} x^2 q_{2m}(x, b_1)\, e^{-b_1^2x^2}\, dx. \tag{78}$$

Here, we have suppressed the dependence of b_1. Using the recurrence relation for the even ordered polynomials, we derive the following system of first-order differential equations for $v_{2m}(c)$. Let $\theta_m \equiv v_{2m}$, then

$$\theta'_m = -a_{2m+1}a_{2(m+1)}\theta_{m+1} - (a_{2m+1}^2 + a_{2m}^2)\,\theta_m - a_{2m}a_{2m-1}\theta_{m-1},$$
$$m = 1, 2, \ldots. \tag{79}$$

In matrix form, (79) is a first-order linear differential equation for the vector Θ whose components are the $\{\theta_n\}$ ordered in increasing n,

$$\Theta' = -B(b_1)\Theta + F(b_1). \tag{80}$$

$B(b_1)$ is that tridiagonal Jacobi matrix associated with the computation of the Gaussian quadrature rule for $b = b_1$. In fact, its eigenvalues are the abscissas of the Gaussian quadrature rule. The solution of (80),

$$\Theta(c) = e^{-Bc}\Theta(0) + B^{-1}F,$$

can be obtained from a spectral decomposition of B. Therefore, its computation is just a by-product of the Gaussian quadrature calculations once the recurrence coefficients $a_n(b_1)$ are obtained.

In conclusion, a domain decomposition method has been

developed for generating orthogonal polynomials given a Gaussian weight on $(-1, 1)$. The formulation of the computational scheme is directed at securing just the recurrence coefficients and the computational scheme takes advantage of the underlying asymptotic structure of the Rys polynomials as a function of b and n. Thus, the solution technique is effective in the sense that it makes maximal use of the analytic properties of the solution and it can produce results of any desirable accuracy.

ACKNOWLEDGMENTS

The assistance and beneficial discussions rendered by both G. W. Hedstrom and P. Nevai are gratefully acknowledged. I also thank Lila Chase for making the three-dimensional plots. NSF has been instrumental in fostering my acquaintance with P. Nevai. This is an example of an ancillary benefit accrued to a Program Director. Thank you, NSF and the Mathematical Sciences Division. This work is supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, by Lawrence Livermore National Laboratory under Contract Number W-7405-Eng-48.

REFERENCES

1. W. Gautschi, in *Recent Advances in Numerical Analysis*, edited by C. deBoor and G. H. Golub (Academic Press, New York, 1978).
2. M. J. Frisch, M. Head-Gordon, H. B. Schlegel, K. Raghavachari, J. S. Binkley, C. Gonzalez, D. J. Defrees, D. J. Fox, R. A. Whiteside, R. Seeger, C. F. Melius, J. Baker, R. L. Martin, L. R. Kahn, J. J. P. Stewart, E. M. Fluder, S. Topiol, and J. A. Pople, *Gaussian 88* (Gaussian, Inc., Pittsburgh, 1988).
3. R. Shepard, I. Shavitt, R. M. Pitzer, D. C. Comeau, M. Pepper, H. Lischka, P. G. Szalay, R. Ahlrichs, F. B. Brown, and J.-G. Zhao, *Int. J. Quantum Chem.* **S22**, 149 (1988).
4. E. Clementi, *Modern Techniques in Computational Chemistry: MOTECC-89* (ESCOM, Leiden, 1989).
5. R. C. Y. Chin, G. W. Hedstrom, and J. Wong, An efficient method for calculating the Compton scattering kernel and its associated angular moments, in preparation.
6. W. Gautschi, *in E. B. Christoffel–The Influence of his Work in Mathematics and Physical Sciences*, edited by P. L. Butzer and F. Fehér (Birkhäuser, Basel, 1981).
7. R. Piessens and M. Branders, "Tables of Gaussian Quadrature Formulas," Appl. Math. Prog. Div., University of Leuven, Leuven, 1975.
8. H. F. King and M. Dupuis, *J. Comput. Phys.* **21**, 144 (1976).
9. R. C. Y. Chin, G. W. Hedstrom, J. R. McGraw, and F. A. Howes, in *New Computing Environments: Parallel, Vector and Systolic*, edited by A. Wouk (SIAM, Philadelphia, 1986).
10. R. C. Y. Chin, G. W. Hedstrom, J. S. Scroggs, and D. C. Sorensen, in *Adv. Comput. Methods Partial Diff. Eq. VI*, edited by R. Vichnevetsky and R. S. Stepleman (IMACS, 1987).
11. G. W. Hedstrom and F. A. Howes, in *Domain Decomposition Methods*, edited by T. F. Chan, R. Glowinski, J. Periaux, and O. B. Widlund (SIAM, Philadelphia, 1989).
12. G. H. Golub and J. H. Welsch, *Math. Comput.* **23**, 221 (1969).
13. W. Magnus, F. Oberhettinger, and R. P. Soni, *Formulas and Theorems for the Special Functions of Mathematical Physics* (Springer-Verlag, New York, 1966).
14. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, AMS 55 (1964).
15. F. W. J. Olver, *J. Res. Nat. Bureau Stand. B* **71**, 161 (1967).
16. P. van der Cruyssen, *Numer. Math.* **32**, 159 (1979).
17. P. G. Nevai, *Orthogonal Polynomials* (AMS, Rhode Island, 1979), p. 124.
18. P. Nevai, private communications.
19. R. S. Varga, *Matrix Iterative Analysis* (Prentice–Hall, Englewood Cliffs, NJ, 1962).