JOURNAL OF COMPUTATIONAL PHYSICS 60, 97–118 (1985)

The Laguerre Method for Solving Integro-Differential Equations

GORDON P. RAMSEY

Department of Physics, Loyola University of Chicago, Chicago, Illinois 60626

Received September 15, 1983; revised May 25, 1984

The Laguerre method proposed by Furmanski and Petronzio [*Nucl. Phys. B* **195** (1982), 237] is used to solve integro-differential equations found in high energy scattering processes and solid state physics. The method uses properties of Laguerre polynomials to convert the integral to a sum, while the remaining differential equation is solved analytically using an evolution operator approach to avoid numerical approximation errors. The present approach is more general in the evolution operator development and in implementation so that the method may be used in solving a more general class of integro-differential equations. The Laguerre method is shown to be more efficient and accurate than other methods used to solve this type of equation. © 1985 Academic Press, Inc.

I. INTRODUCTION

In various problems of mathematical physics, one encounters integro-differential evolution equations of the form [1–3]

$$\frac{dF(x, t)}{dt} = P(x) \otimes F(x, t) \tag{1.1}$$

where the convolution operation $\otimes$ is defined by

$$P(x) \otimes F(x, t) \equiv \int_x^1 \frac{dy}{y} P\left(\frac{x}{y}\right) F(y). \tag{1.2}$$

As an example, in perturbative quantum chromodynamics (QCD), the leading candidate for a theory of strong interactions, $F(x, t)$ represents a quark or gluon structure function, the evolution parameter t is related to the momentum, x is the fractional momentum of a quark which partakes in the scattering process and $P(x)$ is a probability function. Equation (1.1) represents the evolution of the distribution of non-singlet quarks within a hadron as one probes deeper with a more energetic particle. In solid state physics, Eq. (1.1) is a form of the Boltzmann equation, describing the deceleration of charged particles in solids [3].

Traditionally, two approaches have been used to solve (1.1): the Mellin transform [4–6] and brute-force numerical methods [7, 8]. The Mellin transform

0021-9991/85 $3.00

method suffers from notorious difficulties in inverting the transform to find the solution in terms of $F(x, t)$. Brute-force numerical methods tend to be inefficient and prone to instability, depending upon the approximations used to solve (1.1). A third approach has recently been proposed by Furmanski and Petronzio [1], in which they make an expansion of $F(x, t)$ in terms of Laguerre polynomials to convert the convolution (1.2) into a simple sum. The main objectives of this paper are to discuss why this method is more efficient and accurate than the other methods, to propose a more convenient numerical scheme for implementing the method, and to present examples in high energy and solid state physics, for which the Laguerre method is practical. Our approach also provides a basis for extending the method to solve a larger class of integro-differential equations.

The Laguerre method consists of a variable change from x to $z = -\ln(x)$, followed by an expansion and truncation of $P(z)$ and $F(z, t)$ as a finite series of Laguerre polynomials. The convolution (1.2) reduces to a sum of Laguerre polynomials in z. One can then construct an evolution operator [9] to solve the resulting differential equation without making additional approximations (e.g., Euler, Taylor series or Runge–Kutta). Thus, both the integral and differential operations of (1.1) can be solved in closed form. Numerically, the Laguerre method is more stable and more accurate than other methods in that it allows one to deal with the functions $F(x, t)$ directly rather than with their integral transforms.

In Section II, we present the theoretical background of the Laguerre method to solve (1.1) in QCD. We include development of the method along with derivations of the key parameters used in solving (1.1). Many of the details were not covered in the original reference [1]. In Section III, we outline an efficient numerical procedure for implementing the Laguerre method. The new procedure is easier to implement in many cases and at least as accurate as that proposed by Furmanski and Petronzio. A comparison of solution methods is included with regard to accuracy and stability. We present possible applications of the method in Section IV, which include: deep-inelastic lepton-positron scattering and the Boltzmann equation in solid state physics. We summarize the method and its uses in section V.

II. Theoretical Background

A. *The Evolution Equations*

In QCD, the inclusive lepton distribution in the hard lepton-hadron process: $l + h \to l' +$ anything, has the form of a convolution of a lepton-parton (quark or gluon) cross section with parton densities. The lepton distributions and parton densities are expressed in terms of the variables q^2 and x, where q is the momentum transfer to the parton and x is the fraction of momentum of the struck parton relative to the hadron momentum. If we denote the inclusive lepton distribution in the above process by $\rho^{l'}_{lh}(x, q^2)$ and let p label the QCD partons, we have

$$\rho^{l'}_{lh}(x, q^2) = \sum_p [\hat{\rho}^{l'}_{lp} \otimes p_h(x, q^2)], \tag{2.1}$$

where $\hat{\rho}_{lp}^{l'}$ is the lepton-parton cross section and the convolution is defined by

$$C(x) = A(x) \otimes B(x) \equiv \int_x^1 dy\, y^{-1} A(x/y)\, B(y). \tag{2.2}$$

As we probe deeper into the hadron with particles of higher energy, we expect to reveal internal parton structure. In QCD, this translates into investigating the q^2 evolution of the parton densities, $p_h(x, q^2)$.

In QCD with 3 colors and f flavors of quarks, the cross sections (e.g. $\hat{\rho}_{lp}^{l'}$) are written as power series in the strong coupling parameter $\alpha_s(q_2)$, where

$$\frac{\alpha_s(q^2)}{2\pi} = \frac{2}{\beta_0} [ln(q^2/\Lambda^2)]^{-1} \left[1 - \frac{\beta_1}{\beta_0^2} \frac{ln\, ln(q^2/\Lambda^2)}{ln(q^2/\Lambda^2)} + O\left(\frac{1}{ln^2(q^2/\Lambda^2)}\right)\right]. \tag{2.3}$$

In (2.3), $\beta_0 = 11 - \frac{2}{3} f$, $\beta_1 = 102 - \frac{38}{3} f$ and Λ is a free scale parameter in the theory, which is usually extracted from data. The parton densities, whose q^2 evolution is to be investigated, are expressed in three categories. The valence quarks are the primary constitutents of the hadron and their distribution can be written as $q_v(x, q^2)$. The sea of quarks, $q_s(x, q^2)$, which surround the valence quarks, and gluons, $G(x, q^2)$, which hold the hadron constituents together, complete the hadron structure. It is more convenient to write the evolution equations in terms of the variable $t = -(2/\beta_0) \cdot \ln(\alpha_s(q^2)/\alpha_s(q_0^2))$, where α_s is the strong coupling strength and q_0^2 is an initial value of q^2 at which an initial distribution may be determined from data. The evolution equations can then be written as [2]

$$\frac{d}{dt} q_v(x, t) = P_{qq}(x) \otimes q_v(x, t), \tag{2.4a}$$

$$\frac{d}{dt} \begin{pmatrix} q_s(x, t) \\ G(x, t) \end{pmatrix} = \begin{pmatrix} P_{qq}(x) & P_{qG}(x) \\ P_{Gq}(x) & P_{GG}(x) \end{pmatrix} \otimes \begin{pmatrix} q_s(x, t) \\ G(x, t) \end{pmatrix}, \tag{2.4b}$$

where $P_{ij}(x)$ $(i, j = q, G)$ are the probabilities of finding parton j, having momentum xp, within parton i, having momentum p. These probabilities generally depend upon powers of $\alpha_s(q^2)$, and are given to leading order in $\alpha_s(q^2)$ in Ref. [2]. Equations (2.4a) and (2.4b) are denoted the non-singlet and singlet cases, respectively. Typically, one may solve (2.4) for the parton distributions, subject to initial conditions, and convolute the result with the hard scattering cross sections using (2.1) to predict the inclusive lepton distribution.

At this point, a number of different solution methods have been tried. The most popular approach consists of taking the Mellin transform of (2.4) to convert the convolutions into a simple product [4–6]. The evolution equations are then solved in closed form. Problems arise with this method regardless of how one continues the analysis. Comparing the Mellin moments of the evolution functions with experiments requires data in the complete kinematic range of x, $0 < x < 1$, which is generally not available. Alternately, Mellin inversion of the solution is cumbersome,

even in the simplest case [7]. Numerical solution of (2.4) by brute force is generally inefficient and requires approximations, which yield the method numerically unstable [7, 8]. The Laguerre method [1] is presently the most efficient method that calculates the evolutions directly, so that meaningful comparison can be made to data.

The initial step in solving (2.4) is to convert the convolution to a simple operation in the same fashion as the Mellin transform. To understand the nature of the convolution, we note that the functions $P_{ij}(x)$ and the evolution functions are polynomials in the variable x. The convolution (2.2) of two arbitrary polynomials of x results in a polynomial of x plus terms involving $\ln x$. By changing the variable x to $z = -\ln x$, (2.2) takes the form of Laplace transform convolution,

$$P \otimes F = \int_0^z dy\, P(y)\, F(z-y). \tag{2.5}$$

When one convolutes two polynomials of z in this manner, the result is a polynomial of z without extraneous logarithmic terms. Thus polynomials of z are closed with respect to the convolution (2.5). This indicates that the Laplace transform may be a more natural choice for converting (2.4) to a simpler form. The Laplace transform, defined by

$$\mathscr{L}[f(x)] = F(s) \equiv \int_0^\infty dx \cdot e^{-sx} f(x), \tag{2.6}$$

is similar to the orthogonality condition for Laguerre polynomials,

$$\int_0^\infty dx \cdot e^{-x} L_m(x)\, L_n(x) = \delta_{mn}, \tag{2.7}$$

where δ_{mn} is the Kronecker delta function, corresponding to the conditions: $\delta_{mm} = 1$ and $\delta_{mn} = 0$ $(m \neq n)$. The Laguerre polynomials can be derived from their recurrence relations:

$$(n+1)\, L_{n+1}(z) = (2n+1-z)\, L_n(z) - nL_{n-1}(z), \tag{2.8}$$

where $L_0(z) = 1$ and $L_1(z) = 1 - z$. Since the Laguerre polynomials are orthogonal we can expand the polynomials $P_{ij}(z)$ and the evolution functions as sums of the form

$$P(z) = \sum_{n=0}^{\infty} P_n L_n(z), \tag{2.9a}$$

where

$$P_n = \int_0^\infty dy\, e^{-y} L_n(y)\, P(y). \tag{2.9b}$$

Then, we use the property that [12]

$$L_n(z) \otimes L_m(z) = L_{n+m}(z) - L_{n+m+1}(z), \tag{2.10}$$

to convert the convolutions in (2.4) to simple differences. These equations then take the form of simple first order differential equations, for which we can construct evolution operators to generate solutions.

B. *Construction of the Evolution Operator*

Once the integral (2.2) has been transformed into a sum, the remaining first order differential equation may be solved by constructing an evolution operator [9]. Two advantages to this approach include: elimination for the need of approximate numerical solutions by solving the equation analytically and the ability to extend the method to any class of integro-differential equations solvable by use of an evolution operator.

Given a first order ordinary differential equation with real variables having the form $dx/dt = F(x, t)$, we can define a mapping, E_t, which traces the evolution along the solution curve from an initial point, x_0. The mapping is defined in the evolution variable t by $E_t(x_0) = x(t + t_0)$, so that the solution to the differential equation can be written as $x(t) = E_{t-t_0}(x_0)$. The mapping also has an inverse ($E_t^{-1} = E_{-t}$) and a composite rule, ($E_{s+t}(x_0) = E_s(E_t(x_0))$). Determination of the evolution operator is equivalent to finding the solution to the differential equation.

We can write (2.4) as

$$\frac{dQ(x, t)}{dt} = P(x) \otimes Q(x, t), \tag{2.11}$$

where Q represents $q_v(x, t)$ in the non-singlet case, (2.4a) or the matrix $\binom{q_s(x,t)}{G(x,t)}$ in the singlet case, (2.4b).

Denoting the evolution operator (E_t) for $Q(x, t)$ as $E(x, t)$, then the solution to (2.11) can be written as

$$Q(x, t) = E(x, t) \otimes Q(x, t = 0), \tag{2.12}$$

where $Q(x, t = 0)$ is a suitable initial parametrization of the quark and gluon functions [4]. Substituting (2.12) into (2.11), we obtain the differential equation for the evolution operator

$$\frac{dE(x, t)}{dt} = P(x) \otimes E(x, t). \tag{2.13}$$

We now use the Laguerre method to modify the convolution by changing the variable x to $z = -\ln x$ and expand $P(z)$ and $E(z, t)$ in terms of Laguerre polynomials.

The differential equation for the Laguerre coefficients of $E(z, t)$ is, then,

$$\frac{dE_n(t)}{dt} = \sum_{m=0}^{n} p_{n-m} E_m(t), \tag{2.14}$$

where

$$p_{n-m} = P_{n-m} - P_{n-m-1} \qquad \text{and} \qquad (P_{-1} \equiv 0). \tag{2.15}$$

The Laguerre coefficients P_k, occuring in (2.15), may be calculated if one knows the Mellin transform in x (or Laplace transform in z) of the quark and gluon probabilities P_{ij} $(i, j = q, G)$ [10]. The Laguerre coefficients P_k are found by

$$\left(\frac{1}{1-z}\right) \hat{P} \left(\frac{1}{1-z}\right) = \sum_{k=0}^{\infty} P_k z^k \tag{2.16}$$

for each of the probability functions, P_{ij}, where $\hat{P}$ represents the corresponding Mellin transform. One may then calculate the coefficients p_{n-m} in (2.14) by using (2.15). Since the Laguerre sums for $P(x)$ and $E(x, t)$ converge rapidly, we truncate them to a given value of n. We will solve (2.14) for the evolution operator coefficients, $E_n(t)$, in the non-singlet (2.4a) and singlet (2.4b) cases separately.

In the non-singlet case, (2.14) can be written in matrix form as

$$\frac{d}{dt}\begin{pmatrix} E_0 \\ E_1 \\ E_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} P_0 & 0 & 0 & \cdots \\ P_1 & P_0 & 0 & \cdots \\ P_2 & P_1 & P_0 & \cdots \\ \vdots & & & \end{pmatrix} \cdot \begin{pmatrix} E_0 \\ E_1 \\ E_2 \\ \vdots \end{pmatrix}, \tag{2.17}$$

or in matrix notation: $d\varepsilon(t)/dt = P\varepsilon(t)$. The solution to (2.17) is

$$\varepsilon(t) = e^{Pt}\varepsilon(t=0) = e^{P_0 It} e^{At} \varepsilon(t=0), \tag{2.18}$$

where e^{Pt} corresponds to the evolution operator of (2.17), I is the $n \times n$ identity matrix, and A is the lower diagonal $n \times n$ matrix equal to $P - p_0 I$. The initial value of the matrix ε, $\varepsilon(t=0)$, is chosen to be the column matrix consisting of all 1's, which corresponds to $E(z, 0) = \delta(1-z)$. Since $A^{n+1} = 0$, we can write e^{At} as

$$e^{At} = \sum_{k=0}^{n} \frac{A^k t^k}{k!}. \tag{2.19}$$

We now have the explicit form for the evolution operator coefficients $E_n(t)$ and hence, $E(x, t)$. For convenience, we can construct a recursive relation to evaluate the matrix e^{At}. If we let B_m^k represent the sum of elements of the mth row of A^k, then for each value of n, $B_n^0 = 1$ and

$$B_n^{k+1} = \sum_{i=k}^{n-1} p_{n-i} B_i^k. \tag{2.20}$$

The coefficients $E_n(t)$ for the non-singlet case are then given by [Appendix I]

$$E_n(t) = e^{P0t} \cdot \sum_{k=0}^{n} (B_n^k t^k/k!). \tag{2.21}$$

The corresponding evolution operator is

$$E(z, t) = \sum_{n=0}^{N} E_n(t)\, L_n(z), \tag{2.22}$$

from which the valence quark distributions $q_v(x, t)$ may be found.

The singlet case involves a coupled set of equations written in matrix form (2.4b). The evolution operator for the corresponding coefficients $E_n(t)$ is not a simple exponential term as in (2.18). We can construct the singlet version of the evolution operator, e^{Pt}, from the distinct eigenvalues of the P matrix in (2.4b) [11]. Let us denote the eigenvalues of P by λ_1 and λ_2, with $\Delta\lambda = \lambda_1 - \lambda_2$. Then define the projection matrices Q_1 and Q_2 as

$$Q_1 = (P - \lambda_2 I)/\Delta\lambda \qquad \text{and} \qquad Q_2 = -(P - \lambda_1 I)/\Delta\lambda, \tag{2.23}$$

whereby

$$P = \lambda_1 Q_1 + \lambda_2 Q_2. \tag{2.24}$$

Since Q_1 and Q_2 are orthogonal and idempotent, we can write powers of P as $P^k = \lambda_1^k Q_1 + \lambda_2^k Q_2$. Using (2.19), we can then write

$$e^{Pt} = e^{\lambda_1 t} Q_1 + e^{\lambda_2 t} Q_2, \tag{2.25}$$

from which we construct the singlet evolution operator.

Since Q_1 and Q_2 are orthogonal we can write the evolution matrix as the sum of two parts: $E(z, t) = E_1(z, t) + E_2(z, t)$. Then, using (2.14),

$$\frac{dE_{in}(t)}{dt} = \sum_{p=0}^{n} (\lambda_{i,n-p} - \lambda_{i,n-p-1})\, E_{ip}(t), \tag{2.26}$$

where $\lambda_{i,j}$ is the jth Laguerre coefficient of the ith eigenvalue with $\lambda_{i,-1} \equiv 0$. In (2.26), the $E_{in}(t)$ terms form a column vector (as in 2.17) whose elements are 2×2 matrices. The Laguerre coefficients of the eigenvalues ($\lambda_{1,k}$ and $\lambda_{2,k}$) are found by Eq. (2.16).

If we let $\varepsilon_i(t)$ represent the column vectors $E_{in}(t)$ in (2.26), we may write them in the form

$$\varepsilon_i(t) = e^{\lambda_{i,0} t} e^{D_i t} \varepsilon_i(t = 0), \tag{2.27}$$

where D_i are lower-diagonal matrices of $\lambda_{i,k}$ terms, similar to the matrix A in the non-singlet case. As before, we can derive a recursive relation to evaluate the

matrices $e^{D_i t}$. It can be shown that [11]: $\lambda_{1,0} \equiv \lambda = -28/9$ and $\lambda_{2,0} = 0$ (by momentum conservation), so the Laguerre coefficients are given by

$$E_n(t) = \sum_{k=0}^{n} \frac{t^k}{k!} (A_n^{(k)} + e^{\lambda t} B_n^{(k)}), \tag{2.28}$$

where

$$A_n^0 = Q_{2,n}; \quad B_n^0 = Q_{1,n}; \quad A_n^{(k+1)} = \lambda Q_{1,0} A_n^{(k)} + \sum_{i=k}^{n-1} P_{n-i} A_i^{(k)}$$

and

$$B_n^{(k+1)} = -\lambda Q_{2,0} B_n^{(k)} + \sum_{i=k}^{n-1} P_{n-i} B_i^{(k)}.$$

The evolution matrix for the singlet case is given by (2.22) with $E_n(t)$ given by (2.28).

To summarize the procedure for the singlet case, we first calculate the coefficients p_i in (2.15) for all the singlet elements P_{ij} in (2.4b). Then calculate $A_0^0 = Q_{2,0}$ and $B_0^0 = Q_{1,0}$ from (2.23) and the appropriate Laguerre expansion. By knowing any A_n^k and B_n^k, we generate A_{n+1}^k and B_{n+1}^k by the method outlined in Appendix II. The evolution operator is then found from (2.22) and (2.28).

At this point, one may include higher order contributions to the structure function matrix, $Q(z, t)$. We will discuss this technique in a subsequent section. In the next section we outline the key features of the numerical determination of the non-singlet and singlet structure functions using the Laguerre method.

III. Numerical Calculations

A. *Program Details*

In general, the program may be run on any machine with 128K available RAM (8-bit words) in double presicion real arithmetic. In this section, we will discuss the algorithms used to calculate the evolution matrices and associated parameters for the non-singlet and singlet cases. We discuss the accuracy and convergence of the program in Section B, along with criteria for choosing an optimum truncation point for the sums in (2.9) and (2.22). The results for particular initial quark and gluon distributions are presented in Section *C*, in addition to a comparison of methods for solving (2.4a) and (2.4b).

The same program can be used for solving (2.4a) and (2.4b), with the non-singlet as a special case of the singlet. In the non-singlet case, $B_n^k = 0$, for all k, n and $A_n^0 = 1$ for all n in (2.28). This feature allows one to elliminate the a_n^k and b_n^k terms in (*A*2.9) when calculating non-singlet structure functions. Otherwise the procedure outlined

in Section II is applicable to both cases. When choosing a truncation for the sums in (2.9) and (2.22), one must balance the effects of enhanced convergence and accumulated roundoff error. The number of terms included must be large enough to ensure convergence of the series, but small enough so that roundoff errors in matrix multiplications and iterations do not significantly affect the accuracy. When calculating Laguerre coefficients, as in (2.16), one must find the optimal truncation point to satisfy these criteria. We found that about 15 terms were sufficient to ensure convergence of all terms to within a few percent, while roundoff errors caused erratic behavior of the sums beyond 15 terms. Another criterion for deciding on a value of the number of terms in the sums (N) may be the memory capacity of the machine, since array storage increases as N^2.

It is a straightforward process to calculate e_1 ($\equiv Q_{1,0}$) and e_2 ($\equiv Q_{2,0}$). In the non-singlet case these variables are not explicitly used, but in the numerical procedure they correspond to $e_1 = 0$ and $e_2 = I$, where I is the 2×2 identity matrix. In the singlet case, we use (2.23) and the procedure outlined in Section II to calculate e_1 and e_2 analytically and enter these as inputs to the program.

We used (2.16) to calculate the Laguerre coefficients of the Wilson functions $P_{ij}(z)$, whose explicit form is given in Ref. [11]. To be consistent with the treatment in Ref. [1], the functions occur in the form $zP_{ij}(z)$ in the Altarelli–Parisi equations (2.4), so their Laguerre coefficients were calculated accordingly.

In order to calculate the Laguerre coefficients for the structure functions at q_0^2, we assume the x behavior has the form $F(x) = Ax^l(1-x)^m$, where $l > -1$ and $m \geqslant 0$ [4]. Then the Laguerre coefficients F_i are found by using (2.16) as follows. We expand $(1-x)^m$ in a series

$$(1-x)^m = \sum_{i=0}^{\infty} f_i x^i, \tag{3.1}$$

where the coefficient f_i is obtainable recursively by using $f_0 = 1$, $f_1 = -m$ and $f_i = -f_{i-1}((m-i+1)/i)$. The moment of $F(x)$ is given by

$$\hat{F}(n) = A \int_0^1 dx\, x^{l+n-1}(1-x)^m = A \sum_{k=0}^{\infty} f_k/(k+l+n). \tag{3.2}$$

Then

$$\left(\frac{1}{1-z}\right) \hat{F}\left(\frac{1}{1-z}\right) = \sum_{i=0}^{\infty} z^i \left\{ A \left[\sum_{k=0}^{\infty} f_k \frac{(k+l)^i}{(k+l+1)^{i+1}} \right] \right\}. \tag{3.3}$$

From (2.16) we see that the term in brackets corresponds to the Laguerre coefficients, F_i.

Similar series expansions can be used for the functions $P_{ij}(x)$, provided that they remain analytic over the range of x. The main idea is to expand each $P_{ij}(x)$ as a power series in x:

$$P(x) = \sum_{k=0}^{\infty} a_k x^k. \tag{3.4}$$

Then, from (2.16), the Laguerre coefficients for $P(x)$ are given by

$$P_n = \sum_{k=0}^{\infty} \frac{a_n k^n}{(1+k)^{n+1}}. \tag{3.5}$$

This method is particularly useful when $P(x)$ is complicated or when it contains functions whose moments are difficult to calculate [17]. There is a tradeoff between truncation errors in using moments versus the power series expansion. We have found that the power series method is much more accurate for cases where $P(x)$ contains many terms. This is due to multiple truncations of (2.16), one for each term in $P(x)$, versus a single truncation of the power series in (3.5). Furthermore, one does not have to calculate moments for each term in $P(x)$ to find the Laguerre coefficients. The only conceivable drawback to this method would involve functions whose power series are slowly convergent over the range in x. In many cases, modification can be made to power series coefficients to enhance convergence, such as with $\ln(x)$ in Ref. [12]. We have constructed a data bank of power series expansions for often used functions (logs, polynomials, certain logarithmic integrals). By adding and multiplying the power series expansions, we can calculate the series for complex forms of $P(x)$ in a short time and with highly accurate results. Once the Laguerre coefficients P_n are calculated, we use (2.15) to determine the p_i coefficients contained in the evolution matrix (2.20 and 2.28).

The procedure outlined in the last section is used to determine matrices A_n^k and B_n^k, occurring in the evolution matrix coefficients, $E_n(t)$. [Appendixes I and II]. The determination of the non-singlet matrices is a special case of the singlet matrices. For the non-singlet case, one uses the initial conditions $e_1 \equiv 0$ (matrix) and $e_2 = I$ (the identity matrix). Under these conditions, all B_n^k terms are identically zero, $A_n^0 = 1$ for all $n \geqslant 0$ and the A_n^k terms are multiple sums of the p_i matrices given in (2.15). The initial gluon structure function at q_0^2 is set to zero for the non-singlet while the initial valence quark distribution at q_0^2 is given a suitable parametrization as described above. The Laguerre coefficients of the evolution matrices are calculated using (2.21), then one convolutes the evolution matrix with the quark distribution to obtain the non-singlet structure functions. In the singlet case, we save computer time by explicitly evaluating the matrices a_1^1, b_1^1, A_1^0, B_1^0, A_1^1 and B_1^1. As an accuracy check we can show that

$$A_n^n = (e_2 p_1)^n e_2 \qquad \text{and} \qquad B_n^n = (e_1 p_1)^n e_1 \tag{3.6}$$

for all n, so that comparison can be made to the iterative equations. We will discuss this accuracy in the next subsection.

The final step in computing the leading order evolution, is the convolution of the Laguerre coefficients of the evolution matrices with those of the initial quark and gluon distributions. We proceed in a manner analogous to the determination of the evolution matrices, outlined in (2.13) through (2.15). First, we denote the evolution matrix as E and the initial quark/gluon distribution matrix as F and then expand

both as sums of Laguerre polynomials truncated at a sufficiently large number of terms. The convolution is found by

$$E \otimes F \equiv \sum_{n=0}^{\infty} E_n(t)\, L_n(z) \otimes \sum_{m=0}^{\infty} F_m L_m(z)$$
$$= \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} \varepsilon_{n-k}(t)\, F_k \right) L_n(z), \tag{3.7}$$

where $\varepsilon_{n-k} \equiv E_{n-k} - E_{n-k-1}$. In the last step, we used the convolution property of Laguerre polynomials given in (2.10). Evaluating the term in parentheses in (3.7) involves $n+1$ matrix multiplications, which may contribute to the choice of a truncation point for the sum over n.

Higher order perturbative QCD corrections are introduced into the evolution equation via the $P_{ij}(z)$ functions as powers of $\alpha_s(t)$ [13, 21]. The procedure for including this type of correction consists of convoluting the higher order α_s terms in $P_{ij}(z)$ with the evolution matrix as in (2.13) through (2.15). The terms p_i in (2.15) are modified to include the perturbative corrections. Details of this calculation are shown in Section IV A. Non-perturbative corrections, such as higher twist contributions, modify the structure functions directly [14] and may be included after the leading order evolution has been calculated.

B. *Convergence and Accuracy*

The Laguerre method eliminates numerical errors in integration by converting the convolution to a sum, and in differentiation by use of the evolution operator. The two remaining major sources of error are due to truncation of sums and roundoff in matrix multiplication. Since convolution in the variable $z = -\ln x$ leads naturally to use of Laguerre polynomials, it is no surprise that if we expand the key functions in terms of arbitrary orthogonal polynomials in z, the Laguerre expansion converges most rapidly [1]. Sum truncations occur four times in each of the non-singlet and singlet cases: (1) the Wilson functions P_{ij} in (2.16), (2) the initial quark and gluon distributions (3.3), (3) the evolution functions (2.22) and (4) the final structure function convolutions, (3.7). For the Laguerre coefficients of the Wilson functions and initial quark and gluon distributions, summing terms in the Laguerre series to $n = 8$ is sufficient to generate accuracy to within a fraction of one percent. To determine an appropriate cutoff for the evolution function sums, one must consider a tradeoff between truncation error and roundoff error. Each additional term, $E_n(t)$ requires $2n^2 + 4n + 10$ matrix multiplications, which compounds roundoff error at each stage of iteration, even using double precision arithmetic. We tested the effect of roundoff error by comparison of $E_n(t=0)$ to the identity matrix. For $n \gtrsim 15$ in double precision, roundoff error begins to become significant ($>1\%$). Unless one is limited by usable memory, these calculations should be done in double precision arithmetic. Then, the tradeoff between convergence of the Laguerre series and roundoff error is not so critical. The convergence rate of the

structure function convolution (3.7) is somewhat dependent upon the choice of initial conditions, due to the ability of the truncated Laguerre series in (3.3) to accurately approximate the functions at q_0^2. Using double precision, one may be reasonably assured of convergence of the sum (3.7) to within 1% for $n=8$, regardless of the initial choice of the structure function. The largest errors occur for small t values (<0.20) and large x values ($\gtrsim 0.50$) in both singlet and non-singlet cases.

To summarize, sum truncations occur for determination of the Laguerre coefficients of P_{ij} in (2.16), $F(x, q_0^2)$ in (3.3), $E(t, x)$ in (2.22) and for the determination of $F(x, q^2)$ in (3.7). The relative errors in these truncations at $n=6$ for double precision arithmetic are 5%, 2%, 15% and 10%, respectively, on the average. For $n=8$, however, all of these truncation errors are reduced to well below 1%. Roundoff errors become significant when determining $E_n(t)$, for values of n greater than 15. The choice of n in (2.22), and (3.7) may be determined by the desired accuracy and efficiency, but should be normally between 8 and 15 terms.

C. *Results and Comparison of Methods*

The x dependence for the leading order non-singlet structure functions at fixed q^2 values is shown in Fig. 1 while the singlet quark and gluon functions are shown in Fig. 2 and 3. The initial non-singlet function at $t=0$ was parametrized as $x^{0.7}(1-x)^{2.6}$ [11] and the singlet quark and gluon were given by q_s $(t=0)=(1-x)^7$ and G $(t=0)=(1-x)^5$, respectively. In the non-singlet case,

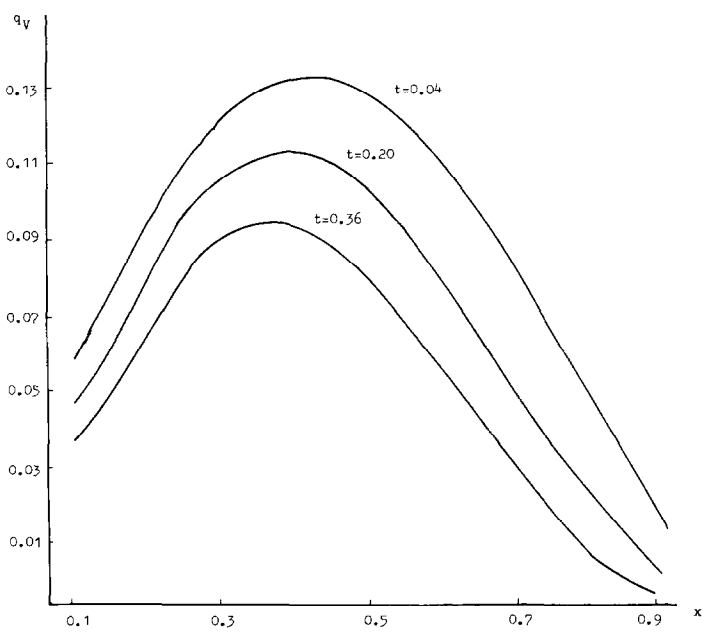

FIG. 1. Valence quark distribution (q_V) vs x for fixed t values.

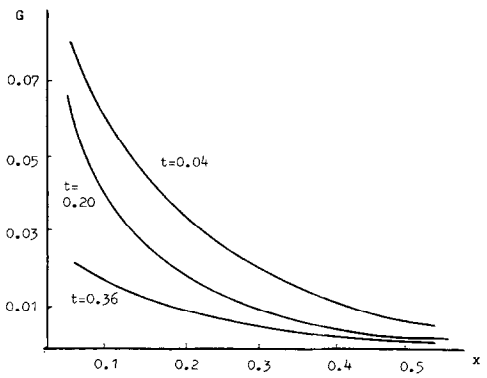

FIG. 2. Gluon distribution vs x for fixed t values.

seven terms in (3.7) were sufficient for convergence to within one part in 10^7 using double precision arithmetic. For the singlet case, eleven terms in (3.7) were necessary to ensure convergence to within 0.5% for most values of x and t, although the functions for large x (>0.5) and t (>0.3) and slightly larger errors. The results in Figs. 1–3 agree favorably with those found by other methods [7, 15, 16].

The major numerical advantage of the Laguerre method is the improvement of run time by a factor of 30 over brute force methods. This factor is relatively independent of the machine, as we tested on an IBM 3033, a PDP-11 and a PRIME 400. Array storage is comparable for the Laguerre method and other "brute force" methods, which evaluate the integrals (2.2) and differentials directly in (2.4). There are many advantages, other than run time improvement, that are realized by the Laguerre method. By expanding the functions in terms of Laguerre polynomials in $(-\ln x)$, we have converted the integrals of (2.2) into sums (2.10), thus avoiding numerical integration errors. Errors in numerical solution of differential equations have been minimized by calculating an evolution operator analytically. The use of a series expansion of the key functions not only provides a

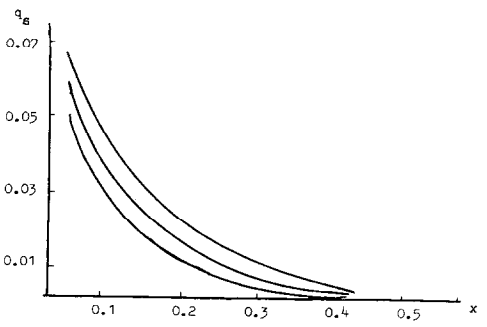

FIG. 3. Sea quark distribution vs x for t values of 0.04 (top), 0.20 (middle), and 0.36 (bottom).

natural starting point for solving (2.4), but allows one the freedom of truncating the program when the desired convergence limit is reached. The only remaining sources of error are truncation and roundoff, which are minimized by the method outlined in the last subsection. Thus, we have gained accuracy, stability and run time without additional memory requirements.

The Laguerre method is also suitable for fitting structure functions to data by suitably adjusting Laguerre coefficients of the evolution functions [1]. Once again, the freedom of choosing the truncations in the sums allows one to have better control of the parameters used in fitting data. Furthermore, since the structure functions of many particle processes (deep-inelastic scattering, lepton pair production and e^+e^-) obey an Altarelli–Parisi type equation (2.4), the Laguerre method can be used in all of their analyses. We will discuss these and other applications of the method in the next section.

IV. Examples Using the Laguerre Method

A. *Deep-Inelastic Scattering*

In Sections II and III we outlined a method for solving the Altarelli–Parisi type equations (2,4) for leading order (QCD), which was oriented toward deep-inelastic scattering. The mathematical technique is also applicable to lepton-pair production processes (Drell–Yan) since the evolution is the same. In deep-inelastic scattering, we can also use the Laguerre method to incorporate higher order perturbative QCD corrections as well as some non-perturbative effects, such as target mass contributions. In this subsection, we outline the process by which these contributions may be added to the deep-inelastic structure functions.

Perturbative QCD corrections to (2.4) occur in the $P_{ij}(x)$ terms as powers of $\alpha_s(q^2)$:

$$P_{ij}(\alpha_s, x) = (\alpha_s/2\pi)\, P_{ij}^{(0)}(x) + (\alpha_s/2\pi)^2\, P_{ij}^{(1)}(x) + \cdots, \tag{4.1}$$

where $P_{ij}^{(0)}(x)$ is the leading order term used in Section II and III. The terms $P_{ij}^{(1)}(x)$ correspond to two-loop corrections to the Feynman diagrams of deep-inelastic processes. These terms have been calculated in $4+\varepsilon$ dimensions using the $\overline{\text{MS}}$ scheme [17] and consist primarily of terms involving powers of x, $\ln x$ and $\ln(1-x)$. To include higher order corrections, one proceeds in the same manner as the leading order calculation [1]. First, one calculates Laguerre coefficients for the $P_{ij}^{(1)}(x)$ terms using the summation method outlined in the last section. Since these involve logarithmic functions and polynomials in x, the summation method is simpler and more accurate than Eq. (2.16). The evolution function coefficients (2.21 and 2.28) are modified according to the revised equations (2.4). Once the new $E_n(x, t)$ are found, the process continues as with the leading order solutions. These solutions are involved and will not be treated here. The Laguerre method makes the problem of including higher order corrections more tractable.

A non-perturbative (with respect to α_s) contribution which must be included in the deep-inelastic analysis is the target mass effect [7]. In the leading order analysis, the mass of the target is neglected. To be kinematically consistent, however, the target mass should be included in the analysis. To leading order in QCD, the method consists of writing the structure function in terms of a variable $\eta(x)$, where

$$\eta(x) = x[(1+\sqrt{1+4M^2/q^2})/(1+\sqrt{1+4M^2x^2/q^2})] \tag{4.2}$$

and M is the target mass. The resulting non-singlet evolution equation has the form [7]

$$\frac{dF(\eta, t)}{dt} = [-12.5 + (50(\ln x)/3)]\, F(\eta, t) + f(x) \otimes F(\eta, t), \tag{4.3}$$

where $f(x) = -[25(1+x^2)]/[3(1-x)]$ and the convolution is defined by (2.2). We solve (4.3) in the same manner as (2.4a) via Eqs. (2.11) through (2.22). First we write the solution of (4.3) in terms of the evolution function $E(x, t)$ as

$$F(\eta, t) = E(x, t) \otimes F(\eta, t=0). \tag{4.4}$$

Substituting (4.4) into (4.3), we obtain the differential equation for $E(x, t)$:

$$\frac{dE(x, t)}{dt} = AE(x, t) + f(x) \otimes E(x, t), \tag{4.5}$$

where $A(x) = -12.5 + [50(\ln x)]/3$. Next, we expand $f(x)$ and $E(x, t)$ in terms of Laguerre polynomials of $z = -\ln x$, and equate the coefficients of each Laguerre polynomial in the sums to get

$$\frac{dE_n(t)}{dt} = AE_n + \sum_{m=0}^{n} (f_{n-m} - f_{n-m-1})\, E_m, \tag{4.6}$$

where f_i and E_i are the Laguerre coefficients of $f(x)$ and $E(x, t)$, respectively. We can write (4.6) in matrix form as in (2.17), where the p_0 terms are replaced by $(f_0 + A)$ and the p_i terms correspond to $(f_i - f_{i-1})$. The corresponding evolution operator coefficients are

$$E_n(t) = \exp[(f_0 + A)\, t] \cdot \sum_{k=0}^{n} A_n^k t^k/k!, \tag{4.7}$$

where $A_n^0 = 1$ and

$$A_n^{k+1} = \sum_{i=k}^{n-1} (f_{n-i} - f_{n-i-1})\, A_i^k. \tag{4.8}$$

The procedure for convoluting $E(z, t)$ with the initial structure function, $F(\eta, t=0)$, is identical to that of the leading order convolution. As before, the results are identical to derivation of the target mass corrected functions calculated using a brute force method [7].

Other non-perturbative corrections to the structure functions (e.g., higher-twist contributions) are included as additional terms to the leading order structure functions [18], and can be added after use of the Laguerre method.

B. *Electron-Position* ($e^- e^+$) *Scattering*

We encounter a slightly different form of Eq. (2.4a) in direct photon production at measured transverse momentum from $e^- e^+$ annihilation [19]. In this process, evolution equation corresponding to (2.4) is given by

$$D_{\gamma/qe}(x, q^2) = D^{\text{Born}}_{\gamma/qe}(x, q_0^2) + \frac{\alpha_s(q^2)}{2\pi} \cdot P_{qq}(x) \otimes D_{\gamma/qe}(x, q^2,), \tag{4.9}$$

where

$$D^{\text{Born}}_{\gamma/qe}(x, q_0^2) \approx \alpha \cdot \frac{1+(1-x)^2}{x(1-x)} \cdot \frac{\sqrt{s}}{2k_T}, \tag{4.10}$$

α is the fine structure constant, s is the center of mass energy and k_T is the transverse momentum of the photon with respect to the incoming electron momentum. Equation (4.9) differs from (2.4) in the lack of a differential on the left-hand side, which eliminates the need for an evolution operator. The Laguerre method is used in converting the convolution to a sum, whereby the resulting equation is solved algebraically. This not only simplifies the problem solution, but also avoids certain approximations and assumptions which must be made to solve (4.9) using Mellin moments [20]. We begin by writing (4.10) as

$$D^{\text{Born}}_{\gamma/qe}(x, q_0^2) = g(q_0^2)\, h(x), \tag{4.11}$$

where $g(q_0^2)$ is found by integrating (4.10) over k_T and $h(x) = [1+(1-x)^2]/[x(1-x)]$. To leading order for 4 quark flavors,

$$\alpha_s(q^2)/2\pi = 0.24/\ln(q^2/\Lambda^2). \tag{4.12}$$

Thus, expanding $h(x)$ and $D(x, q^2)$ as Laguerre series and equating corresponding Laguerre coefficients, we get

$$D_n(q^2) = g(q_0^2)\, h_n + \frac{0.24}{\ln(q^2/\Lambda^2)} \sum_{k=0}^{n-1} p_{n-k}\, D_k, \tag{4.13}$$

where h_n and D_n are Laguerre coefficients of $h(x)$ and $D(x, q^2)$, p_{n-k} are given by (2.15) for $P_{qq}(x)$, and

$$D_0 = (g(q_0^2)\, h_0)/[1 - (0.24\, p_0/\ln(q^2/\Lambda^2))]. \tag{4.14}$$

We can now calculate $D(q_0^2) = \sum_{n=0}^{N} D_n L_n(z)$ at q_0^2 and increment to higher q^2 by using (4.13) and (4.14).

C. *Application to Solid State Physics*

The Laguerre method has applications in any area where the convolution has the form of (2.2). This is the case in solid state physics with the Boltzmann equation which describes the deceleration of charged particles in solids [3]. The equation relating the density of particles, $f(x, E)$, with energy E after traversing a path length x, can be written as

$$\frac{df(x, E)}{dx} = A(E) \int_0^1 \frac{d\tau}{\tau}\, g\left(x, \frac{E}{\tau}\right) h(\tau) = A(E)[g \otimes h], \tag{4.15}$$

where $A(E)$ is an energy dependent term, $\tau = (1 - T/E)$, T is the energy transfer, $g(x, E/\tau) = f(x, E/\tau) - E^{1-2m}(E/\tau)^{2m-1}\, f(x, E)$, $h(\tau) = \tau^{2m}(1-\tau)^{-(1+m)}$ and m is the power parameter of the cross section. In Ref. [3], the equation is solved using a moment approach. If the Laguerre approach is used, one can solve for the x evolution of the density function $f(x, E)$ directly for a fixed energy E. The solution proceeds similarly to that of (2.4a) outlined in Section II. In this problem, the p_i terms of (2.15) are replaced by Laguerre coefficients of $h(\tau)$, which are easily calculable. The Laguerre coefficients of the evolution function are determined by

$$E_n(x) = e^{h_0 x} \sum_{i=k}^{n-1} C_n^k x^k/k!, \qquad \text{where } C_n^0 = 1, \tag{4.16}$$

and

$$C_n^{k+1} = \sum_{i=k}^{n-1} (h_{n-i} - h_{n-i-1})\, C_i^k, \qquad h_{-1} \equiv 0. \tag{4.17}$$

The result from (4.16) can be convoluted with a suitable initial condition for g $(x = 0, E/\tau)$, as outlined in [3], and the density functions, $f(x, E)$, are found using (3.7).

V. Conclusion

We have considered integro-differential equations having the general form

$$\frac{dF(x, t)}{dt} = P(x) \otimes F(x, t), \tag{5.1}$$

where the convolution on the right side is given by (2.2), and x, and t are independent variables. Two common methods used in solving (5.1) consist of numerical solution by brute force (e.g., Euler method with Simpson integration) and use of Mellin transforms. Brute-force solutions require more CPU time and involve approximation methods which are inherently less accurate than the Laguerre method. Mellin transforms are often difficult to invert and suffer in not directly involving the functions we wish to determine. In order to use Mellin transforms effectively in high-energy physics, one requires scattering data to be given in the full range of the variable x, which is generally not available. In other areas of physics, it is often difficult to attach a physical meaning to the Mellin transform of a function. The Laguerre method overcomes all of these disadvantages while allowing one freedom to choose the degree of accuracy by choosing the cutoff points for the sums in Eqs. (2.16), (2.22), (3.3) and (3.7). The only limitations are roundoff error (which is minimized by using double precision arithmetic) and truncation error.

The first step in implementing the Laguerre method is to eliminate the logarithmic terms one obtains when performing the convolution in Eq. (5.1). By writing the polynomials $P(x)$ and $F(x, t)$ in terms of the variable $z = -\ln x$, the convolution results in a polynomial in z. When the convolution (2.2) is written in terms of z, it takes the form of a Laplace convolution with an exponential kernel. This form motivates the expansion of $P(z)$ and $F(z, t)$ in terms of Laguerre polynomials, whose orthogonality condition has the exponential kernel. The convolution of two Laguerrre polynomials in z is a simple sum (2.10), so that the integro-differential equation (5.1) is reduced to an ordinary differential equation. In order to avoid numerical approximations to solve the resulting equation, we analytically find an evolution operator in the variable t which determines the solution based upon an initial parametrization of $F(x, t)$ at $t = 0$.

The use of Laguerre polynomials in z for converting the convolution to a simple operation appears to be the most efficient expansion in terms of convergence over other orthogonal polynomials (e.g., Legendre and Chebyshev) [1]. The closure of (2.2) with respect to z and orthogonality property of Laguerre polynomials contribute to this efficiency. Furthermore, the natural relation between the convolution in z and the orthogonality condition for $L_n(z)$ allows the Laguerre coefficients of a function to be calculated in a straightforward manner (Eqs. (2.16) and (3.5)). The rapid convergence of the Laguerre sums provides numerical stability and efficiency as we showed in Section IIIB.

In this work, we have implemented a method proposed by Furmanski and Petronzio [1] for solving integro-differential equations of the form given by (5.1). We have also introduced a new procedure for finding Laguerre coefficients (Section IIIA), which is simple and accurate for most analytic functions. The approach to determination of the evolution operator was different in this paper, but led to the same result, indicating that this method may be applicable to a wider class of integro-differential equations. The accuracy and efficiency of the Laguerre method was shown to be superior to other methods which have been used to solve Eq. (5.1). We have further illustrated the implementation of this approach to high energy

scattering processes as well as a solid state physics problem (Section IV). Integro-differential equations of the type given by (5.1) are found in many areas of mathematical physiscs, and it is evident that the Laguerre method provides a simple, efficient, and numerically stable way to find solutions without the use of Mellin transforms.

APPENDIX I

In this section, we derive the recursion relation (2.20) and the corresponding form for $E_n(t)$, Eq. (2.21). Each element of the column vector in (2.17) can be written as

$$E_m(t) = e^{p_0 t} \sum_p (e^{At})_{mp}. \tag{A1.1}$$

Using Eq. (2.19), the sum in (A1.1) can be evaluated in powers of A, so that

$$E_m(t) = e^{p_0 t} \sum_{k=0}^{n} \frac{t^k}{k!} \left[\sum_p (A^k)_{mp} \right]. \tag{A1.2}$$

The term in brackets is just the sum of the mth row of A^k, so the problem of finding the elements $E_m(t)$ reduces to evaluating this term. First,

$$B_m^k = \sum_p (A^k)_{mp}, \tag{A1.3}$$

then note that $B_m^0 = 1$ for all m and $B_m^1 = \sum_{i=1}^{m} p_i$, with the p_i given by (2.15). To determine B_m^{k+1} from B_m^k, we write

$$B_m^{k+1} = \sum_{q=0}^{n} \sum_{p=0}^{n} (A)_{mq} \cdot (A^k)_{qp} = \sum_{q=0}^{n} (A)_{mq} \cdot B_q^k. \tag{A1.4}$$

Since A has lower diagonal form, $A_{mq} \neq 0$ only for $m \geq q+1$ and $(A^k)_{qp} \neq 0$ only for $q \geq p+k$. Thus, $B_q^k \neq 0$ for $q \geq k$ and

$$B_m^{k+1} = \sum_{q=k}^{m-1} p_{m-q} B_q^k. \tag{A1.5}$$

Equation (A1.5) with the initial conditions for B_m^0 and B_m^1 allows us to determine B_m^k recursively, with $E_m(t)$ given by

$$E_m(t) = e^{p_0 t} \sum_{k=0}^{n} \frac{t^k}{k!} B_m^k. \tag{A1.6}$$

Equation (A1.5) is equivalent to (2.20) while (A1.6) is identical to (2.21).

APPENDIX II

We derive the sum in (2.28) for $E_n(t)$ in the singlet case and the corresponding recursion relations for A_n^k and B_n^k. In this derivation, the terms $\lambda_{i,k}$ $(i=1, 2)$ are the Laguerre coefficients of the eigenvalues, λ_i. These and the coefficients $Q_{i,k}$ are found by using (2.16). We will consider both $E_{1n}(t)$ and $E_{2n}(t)$ together by writing $E_{in}(t)$, with i representing subscript 1 or 2. Using Eqs. (2.26) and (2.27) with the property that D_i^k has its first $(k-1)$ sub-diagonal elements equal to zero, we can write

$$E_{in}(t) = e^{\lambda_{i,0}t} \sum_{k=0}^{n} \frac{t^k}{k!} \cdot \sum_{p=0}^{n-k} (D_i^k)_{np} Q_{i,p}, \tag{A2.1}$$

where D_i^k is the sub-diagonal matrix

$$D_i^k = \begin{pmatrix} 0 & 0 & \cdots \\ (\lambda_{i,1} - \lambda_{i,0}) & 0 & \cdots \\ (\lambda_{i,2} - \lambda_{i,1}) & (\lambda_{i,1} - \lambda_{i,0}) & \\ \vdots & \vdots & \ddots \end{pmatrix}^k. \tag{A2.2}$$

In our notation, we let

$$\begin{aligned} \sum_{p=0}^{n-k} (D_i^k)_{np} Q_{i,p} &= A_n^k \qquad \text{for } i=2 \\ &= B_n^k \qquad \text{for } i=1 \end{aligned} \tag{A2.3}$$

to obtain Eq. (2.28).

In order to determine A_n^k and B_n^k from A_n^0 and B_n^0, we first consider Eq. (2.14) for finding the Laguerre coefficients, $E_n(t)$:

$$\frac{dE_{in}(t)}{dt} = \sum_{k=0}^{n} p_{n-k} E_{ik}(t). \tag{A2.4}$$

For $i=1$, we substitute (A2.1) into (A2.4) to get

$$e^{\lambda_{i,0}t} \sum_{k=0}^{n} \frac{t^k}{k!} (\lambda_{i,0} B_n^k + B_n^{k+1}) = e^{\lambda_{i,0}t} \sum_{k=0}^{n} \frac{t^k}{k!} \sum_{l=k}^{n} p_{n-l} B_l^k. \tag{A2.5}$$

Equating powers of t and using Eqs (2.23) and (2.24), we get

$$B_n^{k+1} = -\lambda_{1,0} Q_{2,0} B_n^k + \sum_{l=k}^{n-1} p_{n-l} B_l^k. \tag{A2.6}$$

Similarly, for A_n^{k+1}, we obtain

$$A_n^{k+1} = \lambda_{1,0} Q_{1,0} A_n^k + \sum_{l=k}^{n-1} p_{n-l} A_l^k. \tag{A2.7}$$

The remaining task at this point is the determination of A_n^0 and B_n^0, which are equal to $Q_{2,0}$ and $Q_{1,0}$, respectively. One approach involves calculation of $\hat{Q}_i$ directly, which is tedious. Alternatively, if we note that (A2.6) would be a simple recursion without the term involving p_{n-l}, we use the other term to motivate the definition of another variable b_n^k as

$$b_n^k = B_n^k - (-\lambda_{1,0} Q_{2,0})^k B_n^0, \tag{A2.8}$$

with $k > 0$ and $b_0^0 = 0$. Determining the recursion relation for b_n^k is equivalent to finding B_n^k from B_n^0. From (A2.8) and (A2.6) we see that

$$b_n^{k+1} + \lambda_{1,0} Q_{2,0} b_n^k = B_n^{k+1} + \lambda_{1,0} Q_{2,0} B_n^k = \sum_{i=k}^{n-1} p_{n-i} B_i^k, \tag{A2.9}$$

indicating that the b_n^k terms have the same recursion relation, except that $b_0^0 = 0$, while $B_0^0 = Q_{1,0} \neq 0$. A condition analogous to (A2.9) allows one to calculate a_n^k in terms of the A_i^k. Now we relate B_n^0 and A_n^0 to the b_n^k and a_n^k terms by the following.

First note that

$$B_n^n \equiv (D_1^k)_{n0} Q_{1,0} = (\lambda_{1,1} - \lambda_{1,0})^n Q_{1,0}$$

and

$$A_n^n = (D_2^k)_{n0} Q_{2,0} = (\lambda_{2,1} - \lambda_{2,0})^n Q_{2,0}, \tag{A2.10}$$

so that $Q_{2,0} B_n^n = Q_{1,0} A_n^n = 0$ by the orthogonality of Q_1 and Q_2. Using (A2.8), (A2.10) and the idempotency of Q_1 and Q_2, we can write

$$Q_{2,0} b_n^n = -(-\lambda_{1,0})^n Q_{2,0} B_n^0 \quad \text{and} \quad Q_{1,0} a_n^n = -(-\lambda_{1,0})^n Q_{1,0} A_n^0, \tag{A2.11}$$

which relates A_n^0 and B_n^0 to a_n^n and b_n^n. Furthermore, we have

$$A_n^0 + B_n^0 = Q_{1,n} + Q_{2,n} = I, \tag{A2.12}$$

where I is the 2×2 identity matrix. Thus

$$\begin{aligned} A_n^0 &= Q_{1,0} A_n^0 + Q_{2,0}(I - B_n^0) \\ &= Q_{2,0} - [Q_{1,0} a_n^n - (-1)^n Q_{2,0} b_n^n]/\lambda_{1,0}^n \end{aligned} \tag{A2.13}$$

and

$$\begin{aligned} B_n^0 &= Q_{1,0}(I - A_n^0) + Q_{2,0} B_n^0 \\ &= Q_{1,0} + [Q_{1,0} a_n^n - (-1)^n Q_{2,0} b_n^n]/\lambda_{1,0}^n. \end{aligned} \tag{A2.14}$$

Equations (A2.6) through (A2.9), (A2.13) and (A2.14) are all that is necessary to derive the Laguerre coefficients of the evolution matrix in Eq. (2.28).

ACKNOWLEDGMENTS

I thank P. Johnson of the Illinois Institute of Technology for many useful discussions pertaining to implementation of the Laguerre method, and J. Ralston of Argonne National Laboratory for discussions regarding the photon structure functions in e^+e^- scattering.

REFERENCES

1. W. FURMANSKI AND R. PETRONZIO, *Nucl. Phys B* **195** (1982), 237.
2. G. ALTARELLI AND G. PARISI, *Nucl. Phys. B* **126** (1977), 298.
3. K. B. WINTERBON, *Phys. Rev. Lett.* **50** (1983), 1723.
4. A. J. BURAS AND K. J. F. GAEMERS, *Nucl. Phys. B* **132** (1978), 249.
5. A. GONZALES-ARROYO AND C. LOPEZ, *Nucl. Phys. B* **166** (1980), 429.
6. G. PARISI AND N. SOURLAS, *Nucl. Phys. B* **151** (1979), 421.
7. G. RAMSEY, Ph. D. thesis, Illinois Institute of Technology, 1982.
8. N. CABIBBO AND R. PETRONZIO, *Nucl. Phys. B* **137** (1978), 395.
9. D. K. ARROWSMITH AND C. M. PLACE, "Ordinary Differential Equations," Chapman & Hall, London, 1982.
10. H. GEORGI AND H. D. POLITZER, *Phys. Rev. D* **9** (1974), 416; D. J. Gross and F. Wilczek, *Phys. Rev. D* **9** (1974), 980.
11. A. BURAS, *Rev. Mod. Phys.* **52** (1980), 199.
12. M. ABRAMOWITZ AND I. STEGUN, "Handbook of Mathematical Functions," National Bureau of Standards, Washington D.C., 1972.
13. G. CURCI, W. FURMANSKI, AND R. PETRONZIO, *Nucl. Phys. B* **175** (1980), 27.
14. S. J. BRODSKY, E. L. BERGER, AND G. P. LEPAGE, SLAC-PUB-3027, 1982.
15. L. F. ABBOTT AND R. M. BARNETT, *Ann. Phys.* (N.Y.) **125** (1980), 276.
16. F. EISELE, Invited takl, IV Symposium on Elementary Particle Physics, Kazimierz, May 1981.
17. W. FURMANSKI AND R. PETRONZIO, *Phys. Lett. B* **97** (1980), 437; W. A. BARDEEN, A. J. BURAS, D. W. DUKE, AND T. MUTA, *Phys. Rev. D* **18** (1978), 3998; W. FURMANSKI AND R. PETRONZIO, Ref. TH. 3046-CERN, 1981.
18. J. DREES AND E. MONTGOMERY, CERN-EP/83-41, 1983.
19. J. RALSTON AND B. PIRE, ANL-HEP-PR-83-30, 1983.
20. M. GLÜCK AND E. REYA, DO-TH 83/05, Universität Dortmund, 1983.
21. A. DEVOTO, D. W. DUKE, J. D. KIMEL, AND G. A. SOWELL, Florida State preprint, April 1984.