Rose-Hulman Institute of Technology

Rose-Hulman Scholar

Mathematical Sciences Technical Reports (MSTR)

Mathematics

5-1995

Population Genetics: Estimation of Distributions through Systems of Non-Linear Differential Equations

Nacer E. Abrouk
Rose-Hulman Institute of Technology

Robert J. Lopez
Rose-Hulman Institute of Technology

Follow this and additional works at: https://scholar.rose-hulman.edu/math_mstr

Part of the Applied Mathematics Commons, Dynamical Systems Commons, and the Probability Commons

POPULATION GENETICS: ESTIMATION OF DISTRIBUTIONS THROUGH SYSTEMS OF NON-LINEAR DIFFERENTIAL EQUATIONS

Nacer E. Abrouk and Robert J. Lopez

May 1995

Department of Mathematics
Rose-Hulman Institute of Technology
Terre Haute, IN 47803

FAX(812) 877-3198   Phone: (812) 877-8391

Population Genetics: Estimation of Distributions Through Systems of Non-linear Differential Equations

Nacer E. Abrouk and Robert J. Lopez *

Introduction

In stochastic population genetics, the fundamental quantity used for describing the genetic composition of a Mendelian population is the gene frequency. The process of change in the gene frequency is generally modeled as a stochastic process [1] satisfying the stochastic differential equation $dY_t = a(Y_t)dt + \sigma(Y_t)dB_t$. The coefficients involved in this equation (the drift a and the diffusion σ) reflect such mechanisms as mutation, selection, and migration that affect the population. Except in very simple cases, it is difficult to determine the probability law of the stochastic process of change in gene frequency Y_t. Hence, numerically approximating the probability law of Y_t, for $t > 0$, is an important task. We shall present a method for obtaining such approximations, enabling us to study models more realistic than those treated previously. Called the Gauss Galerkin method for its combining of elements of Gauss quadratures and Galerkin approximation, this method was originally proposed by Dawson [2] and then refined by HajJafar [3] and by Abrouk [4]. A Maple implementation of this Gauss Galerkin method is illustrated in this article.

Formulation

Let us assume that a pair of alleles A and a are segregating in a population of large size. By "allele" we mean a form of a gene to which is ascribed the role of transmitting an inherited characteristic. We shall assume that the process of change in gene frequency satisfies a stochastic initial value problem of the form:

$$dY_t = a(Y_t)dt + \sigma(Y_t)dB_t, \tag{1}$$
$$Y_0 = X, \tag{2}$$

where X is a given random variable, B_t is the standard Brownian motion, and the coefficients a and σ satisfy the standard conditions for the existence and uniqueness of the solution. For more details see [5].

We shall assume $a(x) = sx(1-x)(h+(1-2h)x) - ux + v(1-x)$ and $\sigma(x) = \sqrt{x(1-x)/2}$, where s, u, v, and h are given parameters. By applying Ito's formula (see Ethier and Kurtz [5]) to Eq. (1) we get

$$d\phi(Y_t) = (a(Y_t)\phi'(Y_t) + \frac{1}{2}\sigma^2(Y_t)\phi''(Y_t))dt + \sigma(Y_t)dB_t, \tag{3}$$

where $\phi \in C^2(\mathbf{R})$. Eq. (3) is equivalent (by the definition of stochastic integrals) to

$$\phi(Y_t) - \phi(Y_0) = \int_0^t (a(Y_s)\phi'(Y_s) + \frac{1}{2}\sigma^2(Y_s)\phi''(Y_s))ds + \int_0^t \sigma(Y_s)dB_s. \tag{4}$$

*Department of Mathematics and Statistics, Rose-Hulman Institute of Technology.

We specialize Eq. (4) to the function $\phi_k(y) = y^k$, where $k \in \mathbb{N}$, to obtain

$$(Y_t)^k - (Y_0)^k = \int_0^t (a(Y_s)k(Y_s)^{k-1} + \frac{1}{2}k(k-1)\sigma^2(Y_s)(Y_s)^{k-2})ds + \int_0^t \sigma(Y_s)dB_s. \quad (5)$$

Taking the expected value of both sides of Eq. (5), and then using Fubini's theorem, we get

$$E((Y_t)^k) - E((Y_0)^k) = \int_0^t E[(a(Y_s)k(Y_s)^{k-1} + \frac{1}{2}k(k-1)\sigma^2(Y_s)(Y_s)^{k-2})]ds, \quad (6)$$

where $E(\int_0^t \sigma(Y_s)dB_s) = 0$, since $\int_0^t \sigma(Y_s)dB_s$ is a martingale with respect to the natural filtration. By differentiating both sides of Eq. (6) with respect to t, we obtain

$$\frac{d}{dt}E((Y_t)^k) = E[(a(Y_t)k(Y_t)^{k-1} + \frac{1}{2}k(k-1)\sigma^2(Y_t)(Y_t)^{k-2})]. \quad (7)$$

Combining Eqs. (7) and (2) yields the following infinite system of ordinary differential equations

$$\frac{dM_k(t)}{dt} = (vk + \theta_k)M_{k-1}(t) + (k(sh - u - v) - \theta_k)M_k(t)$$
$$+ s(1 - 3h)kM_{k+1}(t) - s(1 - 2h)kM_{k+2}(t), \quad (8)$$
$$M_k(0) = E(X)^k, \quad (9)$$

where E is the expected value and $\theta_k = k(k-1)/4$, for $k = 0, 1, 2, \ldots$, etc., and $M_k(t)$ is the kth moment of the random variable Y_t, defined by $M_k(t) = E(Y_t^k) = \int_0^1 y^k dF_t(y)$, and F_t is the distribution function corresponding to Y_t.

When $s = 0$, Eqs. (1) and (2) admit a unique solution given by

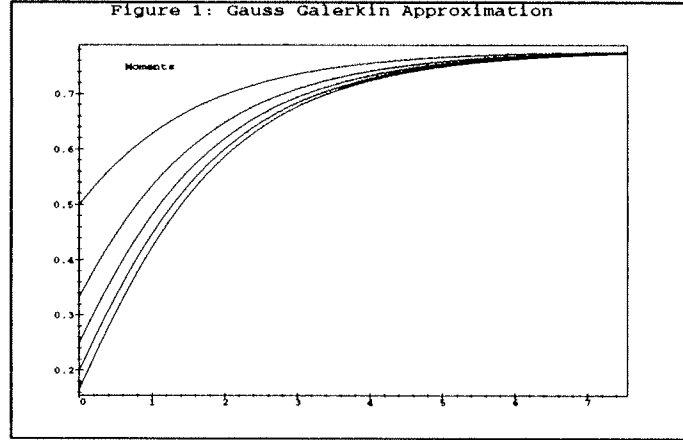$$M_k(t) = [M_k(0) + (kv + \theta_k)\int_0^t M_{k-1}(\eta)e^{(ku+kv+\theta_k+sh)\eta}d\eta]e^{-(ku+kv+\theta_k+sh)t}.$$

When $s \neq 0$, system (1) and (2) form an "open hierarchy" in which the differential equation for M_k contains terms involving M_{k+1} and/or M_{k+2}. Various truncation schemes have been proposed to approximate the solution of system (8) and (9). We shall illustrate the Gauss Galerkin scheme for this model when $s = 2$, $h = 0.5$, and $u = v = 0$. This corresponds to the case where selection occurs ($s = 2$) but there is no mutation or migration ($u = v = 0$). We fix a positive integer n and then retain the first $2n$ equations of system (8) and (9). The resulting finite system is then

$$\frac{dM_k(t)}{dt} = \theta_k M_{k-1}(t) + (k - \theta_k)M_k(t) - kM_{k+1}(t), \quad (10)$$
$$M_k(0) = E(\xi)^k, \quad (11)$$

for $k = 0, 1, 2, \ldots, 2n - 1$. Note that the system given in (10) and (11) involves the $2n + 1$ unknowns $M_0, M_1, \ldots, M_{2n}$, but has only $2n$ equations. One possible closure of this system can be achieved by imposing the condition $\Delta_n = 0$ where Δ_n is the Hankel determinant

$$\Delta_n = \det \begin{pmatrix} M_0 & M_1 & \ldots & M_n \\ M_1 & M_2 & \ldots & M_{n+1} \\ \ldots & \ldots & \ldots & \ldots \\ M_n & M_{n+1} & \ldots & M_{2n} \end{pmatrix}.$$

Figure 1: Gauss Galerkin Approximation

Set $\Delta_n = 0$, and solve for M_{2n} in terms of $M_0, M_1, \ldots, M_{2n-1}$, assuming that $\Delta_{n-1} \neq 0$. The moment M_{2n} is then a rational function $\Phi(M_0, M_1, \ldots, M_{2n-1})$. In the last equation given in (5), replace M_{2n} by Φ to obtain

$$M_k'(t) = (\theta_k)M_{k-1}(t) + (k - \theta_k)M_k(t) - kM_{k+1}(t), (0 \leq k \leq 2n - 2), \tag{12}$$

$$M_{2n-1}'(t) = \theta_{2n-2}M_{2n-2}(t) + (2n - 1 - \theta_{2n-1})M_{2n-1}(t) - (2n - 1)\Phi. \tag{13}$$

For example, if $n = 3$, the function Φ would be

$$\Phi = -\frac{(-M_2M_5 + 2M_3M_4M_5 - M_4 + M_1M_5 - 2M_1M_3M_5 - 2M_1M_4M_2M_5}{(M_2M_4 - M_3^2 - M_1^2M_4 + 2M_1M_3M_2 - M_2^3)}$$
$$+\frac{2M_1M_3M_4^2 + 2M_2^2M_3M_5}{(M_2M_4 - M_3^2 - M_1^2M_4 + 2M_1M_3M_2 - M_2^3)}$$
$$+\frac{M_2^2M_4^2 - 3M_2M_4M_3^2 + M_3^4)}{(M_2M_4 - M_3^2 - M_1^2M_4 + 2M_1M_3M_2 - M_2^3)}$$

Using the initial conditions $M_k(0) = \frac{1}{k+1}$, for $k = 0, 1, 2, \ldots, 5$, we solve the closed system given in (12) and (13) using a fourth-order Runge-Kutta numeric integration. The code for this integration technique was implemented in Maple by Dan Schwalbe and appears in the ODE file in the Maple Share Library. This uses Maple's `evalhf` option to by-pass floating point emulation in favor of direct access to the computing platform's own hardware floating point processor. Maple code that effects this strategy can be found in Appendix A.

Fig. 1 displays Maple graphs of the numerically computed solutions for $\{M_k, 1 \leq k \leq 5\}$ (since $M_0(t) = 1, \forall t$). From the results on which these graphs are based, the probability law of the random variable Y_t, for $t \geq 0$, can then be approximated by a discrete probability measure $\mu_n(t)$ as follows.

The numerical approximation to the solution of the closed system given in (12) and (13) is denoted by the $2n$ functions $\{m_k(t), 0 \leq k \leq 2n - 1\}$. Assuming these $2n$ functions are the first $2n$ moments of a discrete measure (with nodes $\{x_k(t), 1 \leq k \leq n\}$ and weights $\{w_k(t), 1 \leq k \leq n\}$) defined on the interval $[0, 1]$, then

$$m_k(t) = \sum_{j=1}^{n} w_j(t)(x_j(t))^k, \quad 0 \leq k \leq 2n - 1.$$

Since the approximating functions $\{m_k(t), 0 \leq k \leq 2n - 1\}$ are known (at all mesh points), the nodes $\{x_k(t), 1 \leq k \leq n\}$ and the weights $\{w_k(t), 1 \leq k \leq n\}$ can be found (at mesh points) using the numerical algorithm described in the next section.

The Discrete Measure

Our numerical algorithm for finding the nodes and the weights of the Gauss Galerkin measure $\mu_n(t)$ is based on the relationship between the orthogonal polynomials associated with the approximating measure $\mu_n(t)$ and its moments. The nodes $\{x_k(t), 1 \leq k \leq n\}$ and the weights $\{w_k(t), 1 \leq k \leq n\}$ are related to the following real symmetric tridiagonal matrix

$$J_n(t) = \begin{pmatrix} \delta_1 & \gamma_2 & & \\ \gamma_2 & \delta_2 & \ddots & \\ & \ddots & \ddots & \gamma_n \\ & & \gamma_n & \delta_n \end{pmatrix}$$

in such a way that the nodes $\{x_i(n, t), 1 \leq i \leq n\}$ are the eigenvalues of $J_n(t)$ and $\{w_i(n, t), 1 \leq i \leq n\}$ are the squares of the first components of the normalized eigenvectors. The coefficients δ_j, and γ_j, for $1 \leq j \leq n$, are determined by a three-term recurrence for the orthogonal polynomials corresponding to the discrete measure whose nodes and weights are $\{x_k(t), 1 \leq k \leq n\}$ and $\{w_k(t), 1 \leq k \leq n\}$, respectively. Thus, we define

$$p_0(x, t) \equiv 1, \tag{14}$$

$$x p_i(x, t) = \gamma_i(t) p_{i+1}(x, t) + \delta_i(t) p_i(x, t) + \gamma_{i-1} p_{i-1}(x, t), \tag{15}$$

$$\delta_i(t) = \frac{\langle x p_i, p_i \rangle}{\langle p_i, p_i \rangle}, \tag{16}$$

$$\gamma_i^2(t) = \begin{cases} 0 & \text{if } i = 0, \\ \frac{\langle p_{i+1}, p_{i+1} \rangle}{\langle p_i, p_i \rangle} & \text{if } i \geq 1, \end{cases} \tag{17}$$

where $p_{-1}(x, t) \equiv 0$, and $\langle , \rangle$ denotes the inner product with respect to μ_t. For more details see [6]. Note that $p_0(x, t) = m_0(t)$. For each $i = 1, \ldots, n$, Eqs. (16) and (17) are used to obtain δ_i and γ_i before Eq. (15) is used to obtain p_{i+1}. Although these computations are easily carried out in Maple, there is a more sophisticated approach which avoids the computation of the orthogonal polynomials themselves. A careful examination of Eqs. (16) and (17) shows that the inner products on the right hand sides can be expressed in terms of the moments $\{m_k(t), 0 \leq k \leq 2n - 1\}$ computed in (12) and (13). For example, with $p_0 = 1$, we have

$$\delta_1(t) = \frac{\langle x, 1 \rangle}{\langle 1, 1 \rangle} = \int_0^1 x \, d\mu_3(t, x) = m_1(t)$$

and

$$\gamma_1 = 0.$$

For $i = 1$ we have

$$\delta_2(t) = \frac{\langle x p_1, p_1 \rangle}{\langle p_1, p_1 \rangle} = \frac{\int_0^1 x(x - m_1)^2 d\mu_3(t, x)}{m_2(t) - m_1^2(t)} = \frac{m_3(t) - 2m_1(t)m_2(t) + m_1^3(t)}{m_2(t) - m_1^2(t)}$$

and

$$\gamma_2^2(t) = \frac{\langle p_1, p_1 \rangle}{\langle 1, 1 \rangle} = \int_0^1 (p_1(x,t))^2 d\mu_4(t,x) = \int_0^1 (x - \delta_1)^2 d\mu_3(t,x) =$$

$$\int_0^1 (x - m_1)^2 d\mu_3(t,x) = \int_0^1 (x^2 - 2xm_1 + m_1^2) d\mu_3(t,x) = m_2(t) - m_1^2(t).$$

In fact, it can be shown [7] that for $j = 1, 2, \ldots, n$

$$\delta_j = \frac{\Gamma_{j-1}}{\Delta_{j-1}} - \frac{\Gamma_{j-2}}{\Delta_{j-2}} \tag{18}$$

and

$$\gamma_j = \frac{\sqrt{\Delta_{j-2} \Delta_j}}{\Delta_{j-1}}, \tag{19}$$

where

$$\Delta_j = \det \begin{pmatrix} m_0 & m_1 & \ldots & m_j \\ m_1 & m_2 & \ldots & m_{j+1} \\ \ldots & \ldots & \ldots & \ldots \\ m_j & m_{j+1} & \ldots & m_{2j} \end{pmatrix}, \quad j = 0, 1, \ldots, n, \tag{20}$$

and

$$\Gamma_j = \det \begin{pmatrix} m_0 & m_1 & \ldots & m_{j-1} & m_{j+1} \\ m_1 & m_2 & \ldots & m_j & m_{j+2} \\ \ldots & \ldots & \ldots & \ldots \\ m_j & m_{j+1} & \ldots & m_{2j-1} & m_{2j+1} \end{pmatrix}, \quad j = 1, \ldots, n. \tag{21}$$

It is enlightening to see that Eqs. (18)-(21) do, indeed, reproduce the results of (14)-(17). For example,
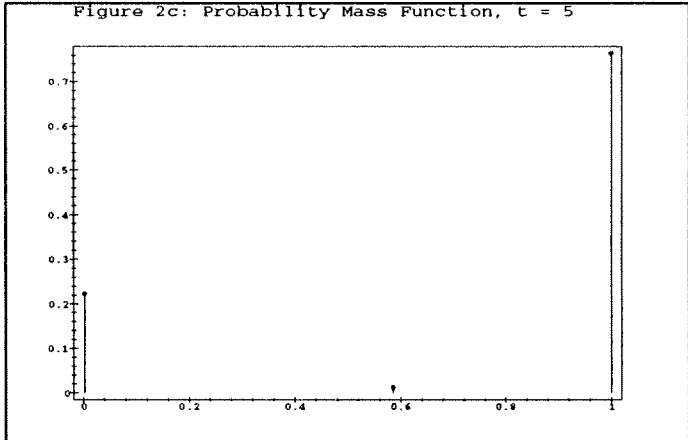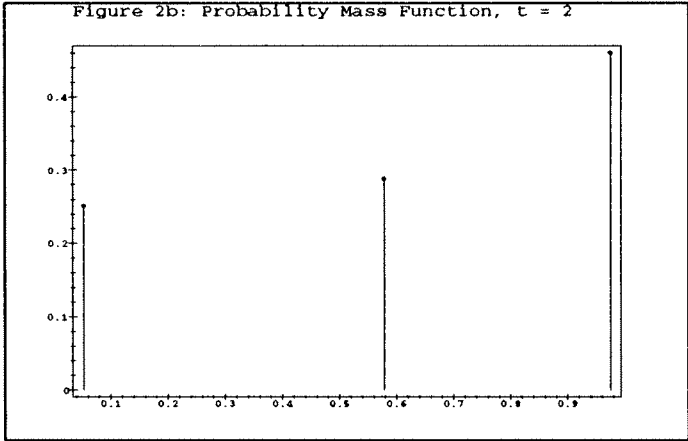
$$\delta_2 = \frac{m3 - m2 \, m1}{m2 - m1^2} - m1$$

and

$$\gamma_2 = \frac{\sqrt{m2 \, m4 - m3^2 - m1^2 m4 + 2 \, m1 \, m2 \, m3 - m2^3}}{m2 - m1^2}$$

Figs. 2a and 2b illustrate the approximate discrete probability measures generated by the matrix $J_n(t)$ for the case $n = 3$, $t = 0$ and 2, respectively. Fig. 2c is a display of the approximate discrete probability measure for $n = 3$ and $t = 5$. In fact, the Maple algorithm in Appendix B produces

$$\begin{bmatrix} 0.5000000000000000 & 0.2886751345 & 0 \\ 0.2886751345 & 0.5000000000 & 0.2581988946 \\ 0 & 0.2581988946 & 0.499999935 \end{bmatrix}$$

$$\begin{bmatrix} 0.6984717770468650 & 0.4002726089 & 0 \\ 0.4002726089 & 0.3428786100 & 0.2028781506 \\ 0 & 0.2028781506 & 0.565164891 \end{bmatrix}$$

Figure 2a: Probability Mass Function, t = 0

Figure 2b: Probability Mass Function, t = 2

Figure 2c: Probability Mass Function, t = 5
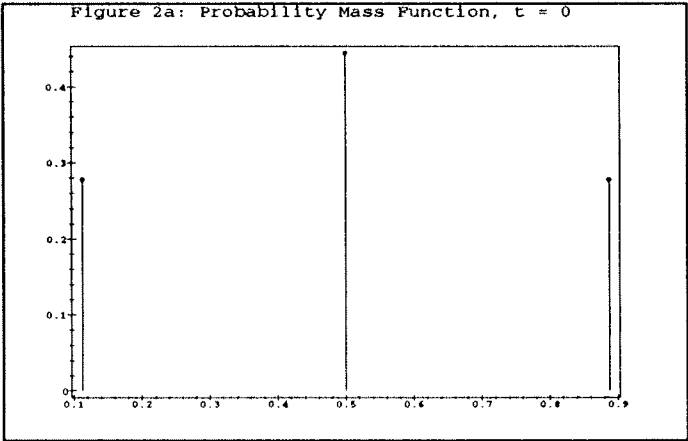
and

$$\begin{bmatrix} 0.7662097131759597 & 0.4141285269 & 0 \\ 0.4141285269 & 0.2429148098 & 0.08808826738 \\ 0 & 0.08808826738 & 0.580974766 \end{bmatrix}$$

for the matrices $J_3(0)$, $J_3(2)$, $J_3(5)$, respectively.

This means the nodes $\{x_i(3,t), 1 \leq i \leq n\}$ are

Table 1: APPROXIMATE NODES			
t	$x_1(3,t)$	$x_2(3,t)$	$x_3(3,t)$
0.00	0.8872983174	0.4999999403	0.1127016343
2.00	0.9897888905	0.5891990080	0.0275273214
5.00	0.5869400430	0.0043482704	0.9988105883

and the corresponding weights $\{w_i(3,t), 1 \leq i \leq n\}$ are, respectively

Table 2: APPROXIMATE WEIGHTS			
t	$x_1(3,t)$	$x_2(3,t)$	$x_3(3,t)$
0.00	0.2777777419	0.2777778045	0.4444444545
2.00	0.6058389860	0.1566197371	0.2375412774
5.00	0.7521699229	0.0237847312	0.2240453461

Moreover, we can verify that

$$m_k(t) = \sum_{i=1}^{n} w_i(3,t)(x_i(3,t))^k$$

holds for each $m_k(t)$ ($k = 0, 1, 2, \ldots, 5$) computed via Appendix A.

Figs. 2a, 2b and 2c are plots of the approximating discrete probability measure (with three nodes and three corresponding weights) at different values of time t. Fig. 2a corresponds to $t = 0$, the initial state. Note the unimodal shape of the discrete measure. Fig. 2b corresponds to $t = 2$, an intermediate state of the dynamical behavior of the gene frequency Y_t. This plot indicates that the discrete measure evolved to a bimodal shape. Fig. 2c displays the discrete measure close to steady state at $t = 5$. Approximately 25% of the mass accumulated at 0 and 75% accumulated at 1. This steady state behavior, which is important in population genetics research, often cannot be obtained when using classical methods such as numerical schemes based on a discretization of the Fokker-Planck equation. For this reason the Gauss Galerkin method is a valuable advance in the analysis of population genetics models.

We have illustrated a method for analyzing a class of genetic population models. In particular, we have solved one particular model of this family and represented its solution graphically. We close by relating our computed results to the physical system described by this model. The model contained in the differential equation (1) tracks the evolution of the alleles **A** and **a** in a population assumed to undergo no migration or mutation ($u = v = 0$). The parameter s measures "selection," which is roughly the propensity for one allele to gain ascendancy over the other. Our choice of $s = 2$ and $h = 1/2$ means that the uniformity encountered in the Mendelian "smooth pea, wrinkled pea" experiments of elementary biology is not present. Moreover, our model does not track genotypes. It merely tracks the presence of the alleles wherever they may be found. Consequently, when we find the limiting distribution to have

the nodes 0 and 1, with corresponding weights .25 and .75, we have discovered that allele A will either disappear entirely (with probability .25) or completely dominate (with probability .75). We leave it to the reader to explore the case $s = 0$ (with $u = v = 0$) to see that then the uniformity of elementary biology is recovered.

References

[1] M. Kimura, and J. Crow: *An Introduction to Population genetics*, Harper and Row, New York, (1970).

[2] D.A. Dawson: *Galerkin Approximation of Non-linear Markov Processes*, Statistics and Related Topics, Proceedings of the International Symposium on Statistics and Related Topics held in Ottawa, Canada, May 5-7, North Holland, (1980).

[3] A. HajJafar: On the Convergence of the Gauss Galerkin Method for the Density of Some Markov Processes, Ph.D. Thesis, Michigan State University, (1986).

[4] N.E. Abrouk: Some Numerical Methods for Singular Diffusions Arising in Genetics, Ph.D. Thesis, Michigan State University, (1992).

[5] S.N. Ethier and T.J. Kurtz: *Markov Processes, Characterization and Convergence*, J. Wiley and Sons, New York, (1986).

[6] J. Stoer and R. Bulirsh: *Introduction to Numerical Analysis*, Springer-Verlag, New York, (1980).

[7] P.J. Davis and P. Rabinowitz: *Methods of Numerical Integration*, 2nd ed., Academic Press, (1984).

APPENDIX A

```
>    # access ODE code from Share Library and load linear algebra package
>    with(share):
>    readshare(ODE,plots):
>    with(linalg):
>
>    # initializations
>    n:=3:
>    L:=2*n-1:
>    M0:=1:
>
>    # define variables for the moments in Eqs. 12 and 13
>    z:=seq(M.j,j=1..L):
>
>    # form the determinant Delta(n) just before Eq. 12
>    d:=matrix(n+1,n+1,(i,j)->M.(i+j-2)):
>    dd:=det(d):
>
>    # obtain M(2n) in terms of M(0), M(1),..., M(2n-1)
>    M.(2*n):=solve(dd,M.(2*n)):
>
>    # form five differential equations for the M's as in Eqs. 12 and 13
```

```
>   for k to L do e.k:=(1/4)*k*(k-1)*(M.(k-1)-M.k)+k*(M.k-M.(k+1)) od:
>
>   # convert differential equations to functional form
>   for k to L do q.k:=unapply(e.k,t,z) od:
>
>   # numeric solution by ODE code
>   N:=rungekuttahf([seq(q.j,j=1..L)],[0,seq(1/(j+1),j=1..L)],1/20,151):
>
>   # extract numeric values for plotting
>   for j to L do sol.j:=makelist(N,1,j+1) od:
>
>   # create Fig. 1
>   with(plots):
>   f1:=plot({sol.(1..L)}, 't'=0..151/20,color=black,axes=boxed):
>   f2:=textplot([1,.75,'Moments']):
>   display([f2,f1],title='Figure 1: Gauss Galerkin Approximation');
```

APPENDIX B

The following code for generating all quantities needed for Figs. 2a, 2b and 2c presupposes that the definitions established in Appendix A are still operative.

```
>   # form Delta(k) of Eq. 20
>   for k from 0 to n+1 do D.k:=matrix(k+1,k+1,(i,j)->M.(i+j-2)) od:
>   for k from 0 to n do Delta.k:=det(D.k) od:
>
>   # form V(k), the matrices in Eq. 21
>   for k from 0 to n do V.k:=minor(D.(k+1),k+2,k+1) od:
>
>   # form the Gamma(k), the determinants in Eq. 21
>   for k from 0 to n do Gamma.k:=det(V.k) od:
>   Gamma.(-1):=0:
>   Delta.(-1):=1:
>   Gamma.(0):=M.1:
>
>   # form the delta(k) of Eq. 18
>   for k to n do delta.k:=(Gamma.(k-1)/Delta.(k-1))-(Gamma.(k-2)/Delta.(k-2))
>   od:
>
>   # form the gamma(k) of Eq. 19
>   for k to n do gamma.k:=sqrt(Delta.(k-2)*Delta.k)/Delta.(k-1) od:
>
>   # sol(j) was computed in Appendix A
>   # S(r) are sets of equations of the form m(k)=numeric approximation of M(k)
>   # at specific times t
>   for t from 1 to 151 by 20 do S.t:={seq(M.j=sol.j[t][2],j=1..L)} od:
>
>   # form the matrices J(n,t) found just before Eq. 14
>   for t to 151 by 20 do J.t:=subs(S.t,array(1..n,1..n,[[delta.1,gamma.1,0],
>           [gamma.1,delta.2,gamma.2],[0,gamma.2,delta.3]])) od:
>
>   # compute eigenvalues and eigenvectors of the matrices J(n,t)
>   # Maple normalizes eigenvectors computed numerically
>   for t to 151 by 20 do evects.t:=eigenvects(J.t) od:
>
>   # form nodes(t), sequences of eigenvalues of the J(n,t) at fixed t
>   for t to 151 by 20 do nodes.t:=seq(evects.t[i][1],i=1..n) od:
>
```

```
>   # form weights(t), sequence of squares of the first components
>   # of the eigenvectors corresponding to nodes(t)
>   for t to 151 by 20 do weights.t:=seq(evects.t[i][3][1][1]^2,i=1..n) od:
>
>   # form the 5 lists measure(t), each containing a sequence of points of the
>   # form (node,weight)
>   for t to 151 by 20 do measure.t:=[seq([nodes.t[i],weights.t[i]],i=1..n)] od:
>
>   # generate Fig. 2a, 2b and 2c
>   for k to 3 do la.k:=[measure1[k],   [measure1[k][1],0]];
>    lb.k:=[measure21[k], [measure21[k][1],0]];
>    lc.k:=[measure121[k],[measure121[k][1],0]] od:
>   f3:=plot({la1,la2,la3},color=black):
>   f5:=plot({lb1,lb2,lb3},color=black):
>   f7:=plot({lc1,lc2,lc3},color=black):
>   f4:=plot(measure1,style=POINT,symbol=circle,axes=boxed,color=black):
>   f6:=plot(measure21,style=POINT,symbol=circle,axes=boxed,color=black):
>   f8:=plot(measure121,style=POINT,symbol=circle,axes=boxed,color=black):
>   display([f3,f4], title=`Figure 2a: Probability Mass Function, t = 0`);
>   display([f5,f6], title=`Figure 2b: Probability Mass Function, t = 2`);
>   display([f7,f8], title=`Figure 2c: Probability Mass Function, t = 5`);
```