MANAGEMENT SCIENCE
Vol. 22, No. 10, June, 1976
Printed in U.S.A.

A NUMERICAL ALGORITHM FOR RECURSIVELY-DEFINED CONVOLUTION INTEGRALS INVOLVING DISTRIBUTION FUNCTIONS*†

ROBERT CLÉROUX‡ AND DENIS J. McCONALOGUE§

Reliability studies give rise to families of distribution functions $F^{(n)}$ defined recursively by the repeated convolution of a distribution function F with itself according to the scheme

$$F^{(1)}(t) = F(t),$$

$$F^{(n+1)}(t) = \int_0^t F^{(n)}(t-x)F'(x)dx, \qquad n \geqslant 1,$$

where F' is the derivative of F, and is usually given by a p.d.f. f. In particular, many systems characteristics are defined in terms of integrals of the form $\int_0^t P^{(s)}(t-x)Q^{(r)}(x)dx$ where $P^{(s)}$ and $Q^{(r)}$ are the sth and rth members of families generated from distribution functions P and Q, not necessarily distinct.

It is seldom possible or convenient to express the $F^{(n)}$ in analytical form. An algorithm based on cubic spline interpolation is given here for recursively generating continuous numerical approximations to the $F^{(n)}$ in a form which allows them to be convoluted together to provide useful approximation to the second of the above integrals.

Introduction

The algorithm to be described here provides a method for numerically calculating certain convolution integrals which occur frequently in the definitions of systems characteristics of interest in reliability theory, and is intended primarily for use when closed forms for the integrals do not exist or are computationally awkward.

In the notation for convolution integrals where

$$(F*G)(t) \equiv \int_0^t F(t-x)G'(x)dt \tag{1}$$

G' being the derivative of G, the algorithm recursively generates continuous numerical approximations to each member of the family of functions $F^{(n)}$ derived from a distribution function F according to the recursive scheme

$$F^{(1)}(t) = F(t),$$

$$F^{(n+1)}(t) = (F^{(n)}*F)(t), \qquad n \geqslant 1, \tag{2}$$

where $F(0) = 0$ and $F(\infty) = 1$. It is restricted to cases where F' is given by a p.d.f. $f \in C^1[0, \infty)$ defined at all points in its range. In addition, the chosen numerical representation makes it possible to calculate useful approximations to integrals of the form

$$(P^{(s)}*Q^{(r)})(t) \tag{3}$$

* Processed by Professor Marcel F. Neuts, Departmental Editor for Applied Stochastic Models and Associate Editor U. N. Bhat; received April 8, 1974, revised March 21, 1975. This paper has been with the authors 6 months for revision.

† The authors are grateful to the two referees of their first draft for a number of very valuable suggestions which are incorporated in this version.

This work was supported by Defence Research Board, Grant No. 9740 18 and National Research Council of Canada, Grant No. A8321.

‡ University of Montreal.

§ University College, Dublin.

where $P^{(s)}$ and $Q^{(r)}$ are the sth and rth members of families generated from distribution functions P and Q according to (2). In principle, a numerical procedure capable of calculating (3) could also be recursively applied to calculate the $F^{(n)}$ in (2); here however, (2) and (3) are treated differently, since (3) inherently involves numerical differentiation with consequent loss of precision whereas (2) does not when the p.d.f. f is available. The use in (2) of these analytically defined derivatives should reduce the build up of error in the recursive generation of the approximations to the $F^{(n)}$.

The algorithm makes it possible to calculate many systems characteristics arising in reliability theory. An enumeration of familiar examples is unnecessary, but two such characteristics which the algorithm was specifically designed to calculate will be set down explicitly. These are $p[N(t) = n]$, the probability of n renewals in time t, and $K(t)$, the availability coefficient for the renewal process with finite renewal times; these are defined as follows. If the lifetimes of the system being renewed are independently and identically distributed with a distribution function F, if the nonnegligible renewal times are also independently and identically distributed with a distribution function G, and if the lifetimes and the renewal times are independent, then $N(t)$ the number of renewals in the period $(0, t)$ has the distribution

$$p[N(t) = n] = (F^{(n)} * G^{(n-1)})(t) - (F^{(n+1)} * G^{(n)})(t) \tag{4}$$

and the availability coefficient $K(t)$, the probability that the system will be working at time t, is given by

$$K(t) = \bar{F}(t) + (\bar{F} * M)(t) \tag{5}$$

where $\bar{F} = 1 - F$, and $M(t) = \sum_{n=1}^{\infty} (F^{(n)} * G^{(n)})(t)$. It is known, Gnedenko et al. [4], that for large t,

$$K(t) \sim \mu_1/(\mu_1 + \mu_2) \tag{6}$$

where μ_1 and μ_2 are the mean values of F and G.

The distribution functions and associated probability density functions on which the algorithm has been tested are also set down since they too are referred to later. They are: the *exponential* distribution with

$$f(t) = \lambda e^{-\lambda t}, \qquad 0 \leqslant t < \infty, \lambda > 0, \tag{7}$$

which generates the analytically defined family

$$F^{(n)}(t) = 1 - e^{-\lambda t} \sum_{r=0}^{n-1} (\lambda t)^r / r!, \qquad n \geqslant 1, \tag{8}$$

the *Weibull* distribution with density function

$$f(t) = \lambda \alpha t^{\alpha - 1} e^{-\lambda t^{\alpha}}, \qquad 0 \leqslant t < \infty, \lambda > 0, \alpha > 0, \tag{9}$$

for which an analytic form for $F^{(n)}$ does not exist for $n > 1$, the *truncated normal* distribution with density function

$$f(t) = \frac{1}{a\sigma(2\pi)^{1/2}} \cdot \exp\{-(t - \mu)^2/2\sigma^2\}, \qquad 0 \leqslant t < \infty, \tag{10}$$

$$a = \frac{1}{\sigma(2\pi)^{1/2}} \int_0^{\infty} \exp\{-(t - \mu)^2/2\sigma^2\}\, dt, \qquad -\infty < \mu < \infty, 0 < \sigma < \infty,$$

for which an analytic form for $F^{(n)}$ does not exist for $n > 1$, and the *gamma* distribution with density function

$$f(t) = t^{\alpha-1}e^{-t/\beta}/\Gamma(\alpha)\beta^{\alpha}, \qquad 0 \leqslant t < \infty, \alpha > 0, \beta > 0 \tag{11}$$

for which

$$F^{(n)}(t) = (\Gamma(n\alpha))^{-1} \int_0^{t/\beta} x^{n\alpha-1}e^{-x}dx. \tag{12}$$

A number of numerical methods have been reported for calculating the renewal function. Smith and Leadbetter [9] have given a series-expansion method for calculating it when F is the Weibull distribution, Soland [10] a method involving numerical solution of the integral equation form, and Jaquette [6] a method based on the asymptotic expansions of the dominating residues of the Laplace transform of the renewal function. All these methods however, deal with immediate replacement (μ_2 negligible compared with μ_1), and none of them has an obvious extension to the more general cases defined above. The present algorithm was designed primarily to find an expression for the time T_0, after which (6) becomes a useful approximation to (5), in terms of F and G and μ_1 and μ_2. The existing methods are inadequate for this investigation since they start from the assumption that K is identically unity.

Computational Difficulties

The basic difficulty in developing an algorithm to calculate (2) and (3) is to find a continuous numerical representation for the functions $F^{(n)}$ which satisfies three criteria. First the form of (3) requires simultaneous approximation of both the function and its first derivative. Second, since the approximands are distributions, the approximation must be one which preserves positivity and monotonicity. Third, the chosen form must be flexible enough to approximate the functions for all values of n and t. For computational simplicity it is obviously desirable to use some form of polynomial approximation, which must here be piecewise, since a family of bounded monotonic functions cannot be represented over the ranges envisaged by a single polynomial of moderately low degree. Partial range approximation in terms of either Legendre or Chebyshev polynomials is ruled out, since neither yields smooth approximants, i.e. they give a good approximation of the function, but the derivative of the approximation is a poor approximation to the derivative of the function. The only "classical" polynomials yielding smooth interpolants, Davis [2], and preserving the monotonicity and positivity of the data are the Bernstein polynomials, but their slow convergence robs them of practical value.

The scheme used here is the common numerical representation of a function as a set of tabulated values at known equally spaced intervals of the independent variable, with interpolation to supply intermediate values. Lagrangian interpolation involving the passing of a polynomial through a number of points adjacent to the region of interest was rejected since the procedure can generate severe undulations even for a curve of moderately high degree, and, in addition, there is first derivative discontinuity at each point (normally each point of tabulation) where a new set of adjacent collocation points is chosen. Instead, cubic spline interpolation is used, since theoretical considerations and practical experience, Ahlberg et al. [1], Greville [5], suggest that good approximations to both function and derivative could be anticipated from the use of some form of spline interpolation, and it is known, Schoenberg [7] that cubic spline interpolation preserves positivity and monotonicity.

In cubic spline interpolation, a function $F(x)$ with known values $F_j \equiv F(x_j)$ at the distinct monotonically spaced values x_j, $j = 0, 1, \ldots, m$, is approximated by a function $y(x)$ defined piecewise by a separate cubic $y_j(x)$ in each panel $[x_j, x_{j+1}]$, where $y(x_j) = F_j$ and $y \in C^2[x_0, x_m]$. As will be shown in the next section, the condition that y and its first and second derivatives should be continuous across the nodes x_j provides only $m - 2$ equations for m parameters, so that for any set of points (x_j, F_j) there is a family of cubic spline interpolants y, each depending on how this indeterminacy is resolved. It is usual to introduce additional properties of F if available or to make additional assumptions about y. Four methods are of interest here, and the effects of two of them on the resulting spline approximation can be deduced from the following standard result, easily established by integration by parts. If $g(x) \in C^2[x_0, x_m]$ is any function interpolating the points (x_j, F_j), $j = 0, 1, \ldots, m$, and y is any cubic spline interpolant, then

$$\int_{x_0}^{x_m} (g'')^2 dx - \int_{x_0}^{x_m} (y'')^2 dx = \int_{x_0}^{x_m} (g'' - y'')^2 dx + 2[y''(x)(g'(x) - y'(x))]_{x=x_0}^{x=x_m}. \quad (13)$$

The right side of (13) will be nonnegative if the indeterminacy is resolved by setting $y''(x_0) = y''(x_m) = 0$; it will then be zero only if $y \equiv g$. This resolution gives rise to a so-called "natural" cubic spline interpolation which is the "smoothest" of all interpolants with first and second derivative continuity, in the sense that it is the one which minimizes the mean-square value of the second derivative over the range. Because of the close relationship between curvature and second derivative, this means effectively that the natural spline is the curve with minimum undulation. However, F belongs to the class of g, so natural splines will, in general, produce an approximation which is smoother than F; in particular when $F''(x_0)$ or $F''(x_m)$ is not close to zero, the approximation to F and F' will be poor close to the corresponding end of the range.

The right side of (13) will also be nonnegative if g and y have the same first derivatives at x_0 and x_m. If the end derivatives of F are available, they can be used to resolve the indeterminacy by imposing the conditions that $y'(x_0) = F'(x_0)$ and $y'(x_m) = F(x_m)$. The resulting y is now the smoothest member of the subclass of interpolants having first derivatives at the ends equal to $F'(x_0)$ and $F'(x_m)$, and will give good approximation to F and F' over the range for a sufficiently well-behaved F.

A third method which can be used when additional information about F is not available is to impose third derivative continuity across x_1 and x_{m-1}. This usually gives better end approximation than natural splines. A fourth method which gives good approximation is to set $y''(x_0) = F''(x_0)$ and $y''(x_m) = F''(x_m)$, when these derivatives of F are available. However, the effects of these two methods on the resulting approximations are not deducible from (13).

Derivation of the Spline Representation

In theory, $F^{(n)}$ is defined in the range $(0, \infty)$, but in practice, only the behavior of the function close to the origin is of interest. Likewise, the number of terms required to sum associated infinite series as in (5) is small, usually of order 10. The method developed here gives an approximation to $F^{(n)}(t)$ in discrete form as a set of $m + 1$ tabulated values of the function corresponding to its values at $t = jh$, where $j = 0, 1, \ldots, m$, and m and h are chosen so that mh covers the range of interest which will vary with the problem; mh will depend on the mean and variance of f. In the calculations referred to in the next section, the maximum value of mh was 60. Since

$$F^{(1)}(t) = \int_0^t f(x)dx, \quad (14)$$

calculation of the tabulated values for $F^{(1)}(t)$ can be based on the analytical identities

$$F^{(1)}(0) = 0,\ F^{(1)}(jh + h) = F^{(1)}(jh) + \int_{jh}^{jh+h} f(x)dx, \qquad j = 0, 1, \ldots, m-1. \quad (15)$$

The quadrature formula chosen for these partial range integrals will depend on the accuracy required, but a wide choice is available if the density functions f are well-behaved in the range $(0, mh)$. The formula chosen to calculate $F^{(1)}$ at this stage is used later to calculate $F^{(n+1)}$ where the integrand is the product of f and the cubic spline approximation to $F^{(n)}$. Since one factor of the integrand is a cubic the integration formula should, for symmetry, be one which implicitly approximates the whole integrand by a sixth degree polynomial.

For $n > 1$, the values of the product integrand at the points jh and $jh + h$ will have been calculated previous to the integration, so a 5-point Lobatto formula using these values, Davis and Rabinowitz [3], meets the requirement with the minimum number of evaluations (three). In this application, it takes the form

$$\int_{jh}^{jh+h} f(x)dx = \frac{h}{180}\left\{9[f(jh) + f(jh+h)] + 49[f(jh+h_1) + f(jh+h_2)] + 64f\left(jh + \frac{h}{2}\right)\right\} - Ch^9 f^{(8)}(\eta), \quad (16)$$

where $h_1 = h(1 - (3/7)^{1/2})/2$, $h_2 = h(1 + (3/7)^{1/2})/2$, $C = 5.4^3(3!)^4/9.(8!)^3 \approx 7 \times 10^{-10}$ and $jh < \eta < jh + h$.

Calculation of $F^{(n+1)}$, $n \geqslant 1$, requires approximations to the analytic forms

$$F^{(n+1)}(kh) = \sum_{j=0}^{k-1} \int_{jh}^{jh+h} F^{(n)}(x) f(kh - x)dx, \qquad k = 1, \ldots, m,$$

$$F^{(n+1)}(0) = 0, \qquad n \geqslant 0. \quad (17)$$

In the typical integral on the right of (17), $F^{(n)}$ is defined only at jh and $jh + h$, and intermediate values required by the Lobatto formula are given by cubic spline interpolation.

A cubic spline approximation to $F^{(n)}$ over the entire range $(0, mh)$ is a function y defined piecewise by a sequence of m different cubic polynomials y_j, one for each panel. If F_j denotes $F^{(n)}(jh)$ (for simplicity, the dependence of y_j and F_j on n is not indicated), then for $jh \leqslant x \leqslant jh + h$, $F^{(n)}$ is represented by y_j where

$$y_j(jh) = F_j, \quad y_j(jh + h) = F_{j+1}, \quad (18)$$

$$dy_{j-1}(jh)/dx = dy_j(jh)/dx, \quad d^2y_{j-1}(jh)/dx^2 = d^2y_j(jh)/dx^2.$$

In general, y_j and y_{j-1} are different cubics, and the third derivative of y is a step function with discontinuities at the points of tabulation.

The y_j can be constructed as follows. Since y_j is a cubic, d^2y_j/dx^2 varies linearly from $x = jh$ to $x = jh + h$. If, with each point (jh, F_j) is associated a real number C_j, as yet unspecified, which is the value given to the second derivative at the point, then d^2y_j/dx^2 can be written in the Lagrangian form

$$\frac{d^2y_j}{dx^2} = \frac{(jh + h - x)}{h} C_j + \frac{(x - jh)}{h} C_{j+1}, \qquad jh \leqslant x \leqslant jh + h, j = 0, \ldots, m-1. \quad (19)$$

For any arbitrary choice of the C's the fourth condition of (18) is satisfied by (19). Integrating twice, and imposing the first two conditions of (18) gives

$$y_j = \frac{(jh + h - x)^3}{6h} C_j + \frac{(x - jh)^3}{6h} C_{j+1} + A_j(jh + h - x) + A_{j+1}(x - jh) \quad \text{where}$$

$$A_j = F_j/h - hC_j/6. \tag{20}$$

For any choice of the C's, the piecewise curve has continuity and second derivative continuity at (jh, F_j). The first derivative is

$$\frac{dy_j}{dx} = -\frac{(jh + h - x)^2}{2h} C_j + \frac{(x - jh)^2}{2h} C_{j+1} + \frac{F_{j+1} - F_j}{h} - \frac{h}{6}(C_{j+1} - C_j). \tag{21}$$

An arbitrary choice of the C's does not give first derivative continuity at (jh, F_j), and it is simple to verify that the third condition of (18) will be satisfied only if

$$\tfrac{1}{4} C_{j-1} + C_j + \tfrac{1}{4} C_{j+1} = \frac{3}{2h^2}(F_{j-1} - 2F_j + F_{j+1}), \qquad j = 1, 2, \ldots, m - 1. \tag{22}$$

A set of C's giving continuity at all the tabulation points results from a solution of the underdetermined set of simultaneous equations defined by (22). The C's produced by any such solution will be functionals of $F^{(n)}$ and functions of h. Since (22) is not defined for $j = 0$ and $j = m$, two additional equations must be obtained from some other source. The approximation is improved if suitable additional properties of the approximand can be incorporated Here the derivatives of $F^{(n)}$ at the origin are defined analytically, and denoting the first and second by F_0' and F_0'', the values are

$$\begin{aligned} F_0' &= f(0), & n &= 1, \\ &= 0, & n &> 1, \quad \text{and} \end{aligned} \tag{23}$$

$$\begin{aligned} F_0'' &= f'(0), & n &= 1, \\ &= [f(0)]^2, & n &= 2, \\ &= 0, & n &> 2. \end{aligned} \tag{24}$$

The use of natural splines obtainable by setting $C_0 = C_m = 0$ is ruled out by (24), since the resulting approximation to $F^{(n)}$ and $F^{(n)'}$ close to the origin would be poor for $n = 1$ and 2, and the resulting error would be propagated to the values of $F^{(n)}$ foı higher n. C_0 could be set to F_0'', but the form of (2) and (3) requires a good approximation to the derivative close to the origin, so it is better to set $dy_0(0)/dx = F_0'$, which gives the additional equation

$$C_0 + \frac{1}{2} C_1 = \frac{3}{h}\left[\frac{F_1 - F_0}{h} - F_0'\right]. \tag{25}$$

This has been found satisfactory in practice.

At $x = mh$, the derivatives of $F^{(n)}$ are not readily available. It would probably be adequate to set $C_m = 0$, since the second derivative at this end is usually close to zero for moderately large t, and the upper end of the range is comparatively unimportant in the calculation, but a better approximation can be anticipated from the less

restrictive assumption of third derivative continuity across $x = (m-1)h$. This directly defines

$$C_{m-1} = (F_m - 2F_{m-1} + F_{m-2})/h^2 \tag{26}$$

and gives the additional (but not independent) equation

$$C_{m-2} - 2C_{m-1} + C_m = 0. \tag{27}$$

The system is thus reduced to $m-1$ equations for C_0 to C_{m-2}, with C_{m-1} given by (26) and C_m by (27).

It is worth noting that the matrix of the system defined by (22) is strongly diagonally dominant, so that the values of the C's away from the ends of the range are largely independent of the additional end equations.

A computationally convenient arrangement for recursively generating the approximations to the $F^{(n)}$ is discussed in the Appendix.

Numerical Accuracy of the Algorithm

It was anticipated initially that the algorithm would be restricted to small values of n because of an expected progressive decrease in the accuracy of the approximation to $F^{(n)}$ with increasing n. This deterioration was expected to arise from two sources: first from errors in the approximation at the nth step which would cause errors in the values at the $(n+1)$st step, the effect being cumulative from $n = 1$ onwards, and second, the movement of the essential support to the right with increasing n. When the algorithm was tested numerically against the exponential (7) and gamma (11) distributions for which $F^{(n)}$ can be computed exactly, no such progressive deterioration occurred. In a large number of tests for λ in (7) in the range (0.03, 1), and α and β in (11) chosen to give means and standard deviations respectively in the ranges (10, 40) and (1,300) and with h in the range (0.1, 0.5), the following behavior was uniformly found. In going from $F^{(1)}$, calculated directly by (16) to better than 10 decimal places, to $F^{(2)}$ which involves spline interpolation, there is a considerable loss of accuracy, the error at the lower end of the range—where it is greatest—being two or less units in the fifth decimal place; the error decreases with increasing t. However, the absolute error over the range $(0, mh)$ in subsequent approximations to $F^{(n)}$, $n > 2$, never exceeds this level, and for each $n > 1$, the approximation improves with increasing t.

Two factors are operative which may account for this persisting accuracy with increasing n. The first is the well-known smoothing effect of numerical integration due to error cancellation, whereby a poor approximation to the integrand can give a good approximate value for the integral. The increasing accuracy with increasing t is a consequence of this property. The second is that the region which makes the maximum contribution to the error, the region close to the origin where the smoothing effect of integration is minimal, is the region which is progressively annihilated as n increases, with the result that errors in the approximation to $F^{(n)}$ close to the origin are heavily damped in the passage to $F^{(n+1)}$, and so do not propagate.

The algorithm has been extensively used to calculate the probability distribution of the number of renewals $p[N(t) = n]$ (4), and the availability coefficient $K(t)$ (5) for the renewal process with finite renewal time over a wide range of means and variances for the cases where F is in turn the Weibull (9), truncated normal (10) and gamma (11) distributions, and G is the exponential distribution (7). For G, the mean μ_2 is $1/\lambda$ and the variance σ_2^2 is $1/\lambda^2$, and the calculations covered the values $0.0325 \leqslant \lambda \leqslant 1$. Denoting the mean and the variance of F by μ_1 and σ_1^2, the ranges were $10 \leqslant \mu_1 \leqslant 40$ and $1 \leqslant \sigma_1^2 \leqslant 306.25$, the corresponding parameters in (9), (10) and (11) being

obtained by numerical solution of the appropriate nonlinear equations. Calculations were frequently repeated using different step lengths h in the range (0.1, 0.5), and an interval of integration $(0, mh)$ chosen so that $K(t)$ had settled down to its asymptotic value given by (5). In each case it was found that $K(t)$ is an oscillating function about its asymptotic value $\mu_1/(\mu_1 + \mu_2)$ and that the calculated values settle down to this asymptotic value to a precision of better than 3 decimal places in the most unfavorable cases. The number of terms of the sum required to calculate $M(t)$ was usually about 10, though in extreme cases when the coefficient of variation, σ_1/μ_1 was close to zero, this rose to 14, the stopping criterion being that $(F^{(n)} * G^{(n)})(t)$ was effectively zero over the range $(0, mh)$. In all cases, calculations with $h = 0.5$ gave the three characteristics of interest to a precision which was adequate for probability calculations. The results of these calculations will be reported separately in a different context.

A useful error analysis of this algorithm is not practicable. A number of analytical expressions for the upper bounds of $|F - y|$ and $|F' - y'|$ over the total range of the approximation in terms of h and the modulus of continuity of F'' exist, e.g. that of Sharma and Meier [8], but these, while useful in establishing properties of spline interpolation such as its uniform convergence to both function and derivative, cannot be used to estimate the errors in the subsequent convolution integrals. Further, analytical forms for $F^{(n)}$, $n > 1$, do not in general exist. In the absence of such an error analysis, the algorithm has to be validated empirically. It is felt that the reported calculations establish the reliability of the present algorithm over the quoted ranges of μ and σ^2. The calculated values of $K(t)$ are the result of repeated applications of the algorithm, during which significant errors in the approximations to $F^{(n)}$ and $G^{(n)}$ could build up so as to make the final values useless. In always producing values which settle down to the known asymptotic value with increasing t, the algorithm has passed a stringent test. The calculations were done on a CDC 6600 Computer with a word length of between 14 and 15 decimal digits.

Appendix

Computational Details

Equation (17) can now be rewritten as

$$F^{(n+1)}(kh) \approx \sum_{j=0}^{k-1} \int_{jh}^{jh+h} y_j(x) f(kh - x)\,dx, \qquad k = 1, 2, \ldots, m. \tag{28}$$

The value of the typical partial range integral in the summation on the right can be interpreted as the inner product of the vector of y values at the Lobatto nodes of the jth panel ($jh \leqslant x \leqslant jh + h$) and the vector of f values multiplied by Lobatto weights at the nodes of the $(k - j - 1)$th panel, traversed in opposite directions. The structure will be evident if it is written in the form

$$\begin{aligned}\int_{jh}^{jh+h} y_j(x) f(kh - x)\,dx \approx F_j \cdot \left\{ \frac{h}{20} f(k_1 h) \right\} + y_j(jh + h_1) \\ \cdot \left\{ \frac{49h}{180} f(k_1 h - h_1) \right\} + y_j\left(jh + \frac{h}{2}\right) \cdot \left\{ \frac{16h}{45} f\left(k_1 h - \frac{h}{2}\right) \right\} \\ + y_j(jh + h_2) \cdot \left\{ \frac{49h}{180} f(k_1 h - h_2) \right\} + F_{j+1} \cdot \left\{ \frac{h}{20} f(k_1 h - jh) \right\}\end{aligned} \tag{29}$$

where $k_1 = k - j$ and h_1 and h_2 are defined as for (16).

The total sum on the right of (28) can be evaluated as the inner product of the first

$4k + 1$ elements of two vectors $\{u_i\}$ and $\{v_i\}$, $i = 0, 1, \ldots, 4m$, traversed in opposite directions, where the u_i are the values of y at the Lobatto nodes in order of increasing x for the total range ($m + 1$ tabulated values, $3m$ interpolated), and the v_i are weighted values of f at the same points, and in the same order with $f(jh)$, $j \geqslant 1$, weighted by $h/10$ to take account of the double occurrence of values at the intermediate tabulated points jh, $j = 1, 2, \ldots, k - 1$. The weighting of $f(kh)$ is irrelevant, since it is multiplied by $u_0 \equiv F^{(n)}(0) = 0$. With this convention

$$F^{(n+1)}(kh) = \sum_{i=1}^{4k} u_i v_{4k-i}, \qquad k = 1, 2, \ldots, m, \tag{30}$$

$$F^{(n+1)}(0) = 0.$$

An economical layout of the calculations is as follows. The vector $\{v_i\}$ which is the same for all values of n is calculated initially and stored; this calculation also provides $F^{(1)}$ by (15) and (16). To calculate $F^{(n+1)}$, $n \geqslant 1$, the spline approximation to $F^{(n)}$ is used to tabulate $\{u_i\}$, and $F^{(n+1)}$ is calculated using (30). The spline coefficients are then calculated for the $(n + 2)$nd cycle. This procedure minimises the number of function evaluations, but has to be paid for by the $8m + 2$ words of storage for the two vectors. The number of operations required to evaluate (30) is $O(m^2)$, and for the rest of the algorithm, $O(m)$.

The calculation of (3) is effected by analytic panel-by-panel convolution of the continuous spline representations of $P^{(s)}$ and $Q^{(r)}$ using (20) and (21); the convolute will be tabulated at the same points as the functions.

Two points of detail should be added. First, there is a loss of significant figures in calculating the second differences on the right of (22) which suggests the need for double-precision calculations in computers with a short word-length, like the IBM 360. Second, in the calculation of (30), and in the analytic convolution of the spline representations, values close to the origin which are effectively zero can be given very small negative values due to round-off error. Steps should be taken to set them to zero.

Listing of three FORTRAN programs to calculate the spline coefficients of a set of equally spaced data with end-conditions (25) and (26), to generate $F^{(n)}$ and its spline coefficients recursively, and to convolute two spline representations, are available from the authors.

References

1. AHLBERG, J. H., NELSON, E. N. AND WALCH, J. L., *The Theory of Splines and Their Applications*, Academic Press, New York, 1967.
2. DAVIS, P. J., *Interpolation and Approximation*, Blaisdell, Waltham, Mass., 1963.
3. ——— AND RABINOWITZ, P., *Numerical Integration*, Blaisdell, Waltham, Mass., 1967.
4. GNEDENKO, B. V., BELYAYEV, YU, K. AND SOLOVIEV, A. D., *Mathematical Methods of Reliability Theory*, Academic Press, New York, 1969.
5. GREVILLE, T. N. E., *Theory and Application of Spline Functions,* Academic Press, New York, 1969.
6. JAQUETTE, D. L., "Approximations to the Renewal Function $m(t)$," *Opns. Res.*, Vol. 20 (1972), pp. 722–727.
7. SCHOENBERG, J. J., in *Inequalities*, O. Shisha, ed., Academic Press, New York, 1967, pp. 255–291.
8. SHARMA, A. AND MEIER, A., "Degree of Approximation of Spline Interpolation," *J. Math Mech.*, Vol. 15 (1966), pp. 759–767.
9. SMITH, W. L. AND LEADBETTER, M. R., "On the Renewal Function for the Weibull Distribution," *Technometrics*, Vol. 5 (1963), pp. 393–396.
10. SOLAND, R. M., "A Renewal Theoretic Approach to the Estimation of Future Demand for Replacement Parts," *Opns. Res.*, Vol. 16 (1968), pp. 36–51.