Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

I-SMOOTH: ITERATIVELY SMOOTHING PIECEWISE-CONSTANT POISSON-PROCESS RATE FUNCTIONS

Huifen Chen

Department of Industrial and Systems Engineering
Chung-Yuan University
Chung-Li, Taoyuan, 320, TAIWAN

Bruce Schmeiser

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907–2023, USA

ABSTRACT

Piecewise-constant Poisson process rate functions are easy to estimate and provide easy random-process generation. When the true rate function is continuous, however, a piecewise-constant approximation is sometimes unacceptably crude. Given a non-negative piecewise-constant rate function, we discuss SMOOTH (Smoothing via Mean-constrained Optimized-Objective Time Halving), a quadratic optimization formulation that yields a smoother non-negative piecewise-constant rate function having twice as many time intervals, each of half the length. I-SMOOTH (Iterated SMOOTH) iterates the SMOOTH formulation to create a sequence of piecewise-constant rate functions having an asymptotic continuous rate function. We consider two contexts: finite-horizon and cyclic. We develop a sequence of computational simplifications for SMOOTH, moving from numerically minimizing the quadratic objective function, to numerically computing a matrix inverse, to a closed-form matrix inverse obtained as finite sums, to decision variables that are linear combinations of the given rates, and to simple approximations.

1 INTRODUCTION

Poisson processes are a classic family of point-process models. Such models are often used to model random arrivals to a system, with each random point being the time that a customer arrives. For simplicity, we refer to arrivals without implying that arrival processes in time are the only application.

A particular model instance is parameterized by its nonnegative rate function $\{\lambda(t); 0 \leq t < \infty\}$. The number of arrivals N in a time interval $[t_{i-1}, t_i]$ is Poisson with mean

$$\mu = \int_{t_{i-1}}^{t_i} \lambda(t)dt.$$

More specifically, the mass function is

$$f_N(n) = \frac{\mathrm{e}^{-\mu}\mu^n}{n!},$$

for $n = 0, 1, 2, \ldots$ and zero elsewhere. The number of arrivals in non-overlapping time intervals are independent.

In practice the rate function is often specified to be piecewise constant; that is,

$$\lambda(t) = \lambda_i \text{ for } t_{i-1} < t \leq t_i$$

for $i = 1, 2, ..., k$. We assume that $\lambda(t_0) = \lambda_1$.

We consider two contexts for the rate function. First, in the *finite-horizon* context the model's time horizon is finite at t_k, with later times being irrelevant. Second, in the *cyclic* context, the time horizon is defined to time infinity by cycling the rates, so that for any time $t > t_0$ the rate $\lambda(t)$ is used at times $t, t_k + t, 2t_k + t, \ldots$. In the cyclic context, λ_1 follows λ_k, as might happen if $k = 7$ and each time interval

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

corresponds to a day of the week. In the finite-horizon context λ_1 and λ_k are simply the first and last rates, respectively, as might happen when the time intervals are rates estimated into the near-term future.

The piecewise-constant form is often a modeling assumption to approximate a continuous rate function. Two advantages—rate estimation and random-process generation—balance the loss of modeling accuracy. First, piecewise-constant rates are easy to estimate; n_i arrivals in m realizations of the interval $(t_{i-1}, t_i]$ implies an estimate that

$$\lambda(t) = \frac{n_i}{m(t_{i-1} - t_i)}$$

for $t_{i-1} < t \leq t_i$.

Second, arrival times can be generated via the inverse transformation. Given an arrival at time s, the next arrival is at the time t satisfying

$$u = 1 - \exp(-\int_s^t \lambda(v)dv),$$

where u is a uniform (0,1) random number. Equivalently,

$$\int_s^t \lambda(v)dv = -\ln(1-u);$$

that is, the area under the rate function between times s and t is exponential with mean one. For piecewise-constant rate functions, the cumulative rate function $\Lambda(t) = \int_0^t \lambda(v)dv$ is piecewise linear, so solving for time t in terms of the previous arrival time s and the random number u is straightforward, as discussed in Klein and Roberts (1984).

We organize this paper as follows. In Section 2 we state the general problem of smoothing a given piecewise-constant rate function. In Section 3 we discuss five measures of solution quality. In Section 4 we discuss our approach, which is to double the number of intervals each with half of the length; each new pair of intervals has an average rate equal to the original interval. In Section 5 we discuss a simple slope-matching heuristic for determining the new rate for each interval. In Section 6 we state objective functions for the cyclic and finite-horizon contexts. Given k rates, we use these objective functions to determine the $2k$ new rates, in a process that we refer to as SMOOTH (Smoothing via Mean-constrained Optimized-Objective Time Halving), which when applied iteratively is I-SMOOTH. In Section 7 we derive a sequence of solution methods, beginning with numerical solution of the mean-constrained quadratic minimization problem using off-the-shelf software, then a matrix inversion, then a closed-form sum, and approximations obtained by truncating the sums. In Section 8 we present four examples. Finally, we end with a brief discussion of some additional issues, including the number of times, say n, to double the number of intervals.

2 THE PROBLEM

Suppose that we are given the piecewise-constant rates λ_i corresponding to the time intervals $(t_{i-1}, t_i]$ for $i = 1, 2, \ldots, k$. We consider only equal-length time intervals. Without lost of generality, we assume $t_i = i$ for $i = 0, 1, \ldots, k$. We are also given the modeler's horizon context. If finite horizon, then $t_0 \leq t \leq t_k$; if cyclic, then $t_0 \leq t < \infty$.

Our problem is to determine a new Poisson rate function, say τ, that is a "better" rate function than λ, despite λ being the piecewise-constant Poisson rate function specified by the modeler. In this section we discuss possible meanings of "better".

First, we take as a constraint that the new rate function τ should maintain the essence of the specified rate function λ. In particular, we require that for each time interval $i = 1, 2, \ldots, k$ the mean number of arrivals remains unchanged:

$$\int_{t_{i-1}}^{t_i} \tau(t)dt = \int_{t_{i-1}}^{t_i} \lambda(t)dt.$$

We ignore the origin of the given constant rates, which might be statistical estimates or might be part of a designed experiment. Our goal is only to smooth the function over time; any statistical error in the given constant rates will remain. For methods that estimate rate functions based on given data, see Kuhl and Wilson (2000, 2001) and Leemis (1991).

Second, we take as a constraint that the solution should be nonnegative. Nevertheless, we ignore negativity until Section 9.

Third, we take as a constraint that the solution should be symmetric in time, just as Poisson processes are. That is, reversing the subscripts on $\lambda_1, \lambda_2, \ldots, \lambda_k$ should result in reversing the time index of τ.

Our objective is that the new rate function is "smoother" than the piecewise-constant rate function λ, which requires a definition of "smoother". We suppose that the true unknown rate function is continuous and that τ should have less variation than λ, with possibly τ being continuous. Because the true rate function is unknown, and because we don't have data, the specific definition of an objective function is part of the research problem.

We do know, though, that the definition of "smoother" needs to depend upon the rate context. For a cyclic rate function, the value of $\tau(t_0)$ should be close to the value of $\tau(t_k)$. For the finite-horizon context, the difference $\tau(t_0) - \tau(t_k)$ is irrelevant.

3 SOLUTION QUALITY

In addition to the constraints of nonnegativity and maintaining the expected number of arrivals for each of the k intervals, we need criteria for determining the quality of a given solution procedure and for comparing multiple procedures.

The primary measure of a solution procedure is the quality of the smoothed rate function. Ideally, most humans would look at the smoothed function and like it. In a computer implementation, however, some definition of smoothness is needed; our definition is in Section 6.

There are many possible definitions of smoothness. We do not compare them here, but rather argue that ours is intuitive, works well on various examples, and is reasonably easy to compute. If other definitions were advocated, they could be compared by considering a particular *true* continuous rate function, computing the expected number of arrivals for each time interval, providing the resulting expected values as the piecewise-constant rate function λ, and applying the solution procedure to obtain the new rate function τ. The deviation of τ from the true rate function would be a measure of the quality of the solution procedure. Various true functions could be used, with a good procedure being robust to the particular function.

A second measure of a solution procedure is the work required to obtain the solution τ. A simple procedure is better than a complicated procedure. Finding a tractable definition of smoothness is a good start.

A third measure is the difficulty of explaining the method to a potential user, as well as the extent to which the method's explanation provides confidence. Here, again, the definition of smoothness is central.

A fourth measure is the amount of information requested from the user. In an ideal solution procedure, nothing is requested beyond $\lambda_1, \ldots, \lambda_k$ and an indication of whether the rate is finite horizon or cyclic.

A fifth measure is, given the current arrival time s, the ease with which the next pseudo-random arrival time can be generated.

4 APPROACH: DOUBLING THE NUMBER OF INTERVALS

Recall that we are given a Poisson process rate function $\lambda(t)$ in the form of k nonnegative constants $\lambda_1, \lambda_2, \ldots, \lambda_k$ corresponding to the time intervals $(t_{i-1}, t_i]$ for $i = 1, 2, \ldots, k$, where the range of t depends on the context: finite-horizon or cyclic.

Throughout this paper, we pursue the solution approach of returning a piecewise-constant rate function with $2k$ pieces, each of length half that of the original intervals. A primary appeal of this approach is that it can be applied a second time to obtain $4k$ and a third time to obtain $8k$ intervals. In the limit the approach yields a continuous approximation to the original k constant pieces without the need to specify a functional form. For now, however, we consider only the single iteration of creating $2k$ rates given k rates.

The new rate function is

$$\tau(t) = \begin{cases} \lambda_i - \gamma_i & \text{for } t_{i-1} < t \le (t_{i-1}+t_i)/2 \\ \lambda_i + \gamma_i & \text{for } (t_{i-1}+t_i)/2 < t \le t_i \end{cases}$$

for $i = 1,2,\ldots,k$. That is, if the left half of the rate of interval i is decreased by γ_i, then the right half must be increased by γ_i for $i = 1,2,\ldots,k$. Any values of the decision variables $\gamma_1, \gamma_2, \ldots, \gamma_k$ satisfy the k constraints that the expected number of arrivals is unchanged for every interval.

Figure 1 illustrates the use of γ increments to update a sequence of finite-horizon piecewise-constant rates. Two sets of rates are shown: the original six rates $3/4, 5/4, \ldots, 13/4$ and the updated twelve rates $5/8, 7/8, \ldots, 27/8$, which result from the increments $\gamma_i = 1/8$ for $i = 1,2,\ldots,6$. These increment values are intuitively correct, in that constant jumps in the original rates result in constant jumps in the updated rates. Repeatedly applying the same doubling process results in the continuous limiting rate function $\tau(t) = 1/2 + t/2$ for $0 \le t \le 6$, the line that passes through the midpoints of the original piecewise-linear rate segments. In Sections 5 and 6 we develop ideas for choosing γ values in situations where appropriate values are less obvious. Section 8 contains three illustrative examples and one real-world example, with figures.

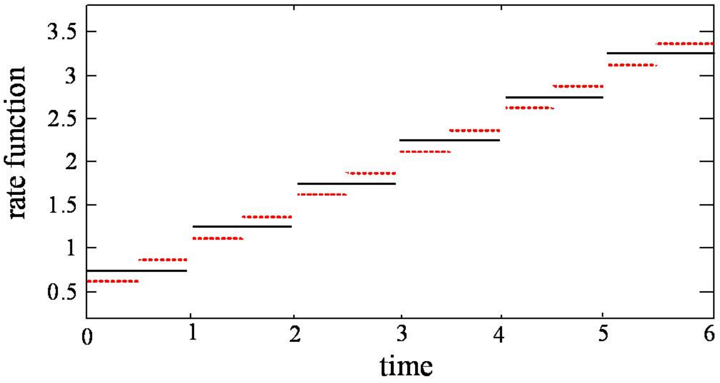

Figure 1: The finite-horizon rates $\lambda_i = 0.25 + i/2$ for $i = 1,2,\ldots,6$ (shown as solid lines) and the twelve rates obtained by doubling the number of intervals one time (shown as the dotted lines). The increments shown are the intuitively correct values $\gamma_i = 1/8$, for $i = 1,2,\ldots,6$.

To obtain non-negative rates, the constraints $|\gamma_i| \le \lambda_i$ for $i = 1,2,\ldots,k$ must be enforced. These constraints are satisfied by the intuitively correct γ increments in Figure 1. How to satisfy non-negativity in general is discussed briefly in Section 9.

5 SLOPE MATCHING

This section is intended to provide motivation and intuition, but is not needed to understand the rest of the paper. Therefore, the reader may wish to skip to the next section.

Before discussing our definition of smoothness in Section 6, we consider a "local" definition of smoothness. The idea is simple: for each interval i, choose the value of γ_i to match the slope of λ near interval i. The definition of slope is defined in terms of first differences, since the first derivative is zero except at the jumps in λ.

In this sense, the slope of the original rate function, λ, at segment i is

$$\frac{\lambda_{i+1} - \lambda_{i-1}}{2},$$

the difference in the λ values adjacent to segment i divided by the time difference between their midpoints.

Similarly, the slope of the new rate function, τ, is

$$\frac{(\lambda_i + \gamma_i) - (\lambda_i - \gamma_i)}{1/2},$$

the difference between the two new halves divided by the time difference between their midpoints.

Setting the two slopes equal yields the first-order slope-matching solution

$$\gamma_i = \frac{\lambda_{i+1} - \lambda_{i-1}}{8}. \tag{1}$$

The end intervals, 1 and k, are treated according to whether the rate context is cyclic or finite horizon. If cyclic, then $\lambda_0 = \lambda_k$ and $\lambda_{k+1} = \lambda_1$. If finite horizon, then the left-end rate is $2\lambda_1 - \lambda_2$ and the right-end rate is $2\lambda_k - \lambda_{k-1}$. that is, the slope is defined using only segments 1 and 2 or $k-1$ and k.

For equal-length intervals, the solution is also that obtained by fitting a quadratic through λ_{i-1}, λ_i, and λ_{i+1} and matching the fitted slope at the midpoint of segment i. This solution can be modified easily to consider unequal-length segments.

The first-order slope-matching solution in Equation (1) can be extended to using multiple rate-value differences. Again we take as motivation that rates that increase by a constant amount should result in a straight line; that is, if $\lambda_{i+1} - \lambda_i = \delta$ for some sequence of intervals, then the increments γ_i should be equal for those intervals.

Consider the lth-order solution obtained by using l differences. Then

$$\gamma_i = \sum_{j=1}^{l} c_j^{(l)} (\lambda_{i+j} - \lambda_{i-j}). \tag{2}$$

From Equation (1), $c_1^{(1)} = 1/8$, the value that yields a straight line. Ignoring end effects, the generalization is, for interval i,

$$\sum_{j=1}^{l} c_j^{(l)} (\lambda_{i+j} - \lambda_{i-j}) = 2\delta/8, \tag{3}$$

which yields

$$\sum_{j=1}^{l} j c_j^{(l)} = 1/8. \tag{4}$$

For $l > 1$, there are many solutions that yield a straight line in the sense of γ_i values being the same for multiple interval i. Of these solutions, we favor those with weights whose magnitudes decrease with lag j. Some thought, maybe developed only with some experimentation, suggests that the weights should alternate signs, with $c_j^{(l)}$ being positive when j is odd. Also based on some experimentation, for $0 < r < 1$ we assume the power series

$$c_{j+1}^{(l)} = -r c_j^{(l)}. \tag{5}$$

The power-series assumption leads to the following weights.

Result 1 For a sequence of rates with constant changes, the unique power-series weights that yield a straight line are

$$c_1^{(l)} = \frac{(1+r)^2}{8[1 - (1 + l + rl)(-r)^l]} \tag{6}$$

and $c_j^{(l)} = (-r)^{j-1} c_1^{(l)}$ for $j = 1, 2, ..., l$.

In Section 7.2, we find that $r = 1/3$ yields an approximation to the solution of the more-precise optimization formulation of Section 7.1.

6 DEFINING SMOOTHNESS

For a continuous function f over the interval $[a,b]$, a well-known definition of smoothness is the integral of squared second derivatives

$$z = \int_a^b \left[\frac{d^2 f(t)}{dt^2}\right]^2 dt. \tag{7}$$

In this definition, a smaller value corresponds to a smoother function. Perfect smoothness is a straight line, which has smoothness $z = 0$.

By analogy, for our piecewise-constant functions defined by $\underline{\lambda}^T \equiv (\lambda_1, \lambda_2, \ldots, \lambda_k)$, we define smoothness by

$$z_F = \sum_{i=1}^{k-2} [\lambda_{i+2} - 2\lambda_{i+1} + \lambda_i]^2 \tag{8}$$

when the problem horizon is finite and by

$$z_C = z_F + (\lambda_{k-1} - 2\lambda_k + \lambda_1)^2 + (\lambda_k - 2\lambda_1 + \lambda_2)^2 \tag{9}$$

when the problem is cyclic. The extra two terms in Z_C arise from the interpretation that λ_k is followed by λ_1. These two terms are irrelevant to Z_F because the given rates are interpreted to terminate at t_k rather than to cycle back to t_0. Finite-horizon rates with constant step sizes $\lambda_{i+1} - \lambda_i$ have perfect (i.e., $z_F = 0$) smoothness. Cyclic rate functions have perfect (i.e. $z_C = 0$) smoothness only when all rates are equal.

Our objective function is to optimize the smoothness of the $2k$ pieces of the new rate function. At the centers of the k original pieces are jumps of size

$$\alpha_i \equiv (\lambda_i + \gamma_i) - (\lambda_i - \gamma_i) = 2\gamma_i$$

for $i = 1, 2, \ldots, k$. Between the k original pieces are jumps of size

$$\beta_i \equiv (\lambda_{i+1} - \gamma_{i+1}) - (\lambda_i + \gamma_i)$$

for $i = 1, 2, \ldots, k-1$; when the problem is cyclic, the kth such jump is

$$\beta_k \equiv (\lambda_1 - \gamma_1) - (\lambda_k + \gamma_k).$$

For notational simplicity, we follow the following convention for subscripts. When the context is cyclic, we define $\lambda_{k+1} \equiv \lambda_1$ and $\gamma_{k+1} \equiv \gamma_1$; more generally, if a subscript is less than 1, then add k; if greater than k, then subtract k. When the context is finite horizon, λ_0 and λ_{k+1} are given constants, specified as initial and final rates to which the smoothed rate function should approach.

The smoothness of the $2k$ pieces is the sum of the squared differences of adjacent jump sizes. Let $\underline{\gamma} \equiv (\gamma_1, \gamma_2, \ldots, \gamma_k)^T$, the vector of decision variables. For the cyclic problem our objective function is to minimize

$$z_C(\underline{\gamma}) \equiv \sum_{i=1}^{k} (\alpha_i - \beta_i)^2 + (\beta_i - \alpha_{i+1})^2,$$

a function of the $2k$ steps.

For the finite-horizon problem, the first and kth rates are not adjacent. We consider four cases, all incorporated into minimizing the objective function

$$z_F(\underline{\gamma}) \equiv \sum_{i=1}^{k-1} (\alpha_i - \beta_i)^2 + (\beta_i - \alpha_{i+1})^2 + I_L[(\lambda_1 - \gamma_1 - \lambda_0) - \alpha_1]^2 + I_R[\alpha_k - (\lambda_{k+1} - \lambda_k - \gamma_k)]^2.$$

The four cases are obtained via the indicator functions I_L and I_R. If $I_L = I_R = 0$, then the smoothed curve considers only the $2(k-1)$ second differences of the $2k$ new rates. If $I_L = 1$ and $I_R = 0$, then the smoothed rate function is asked to link to the given initial rate λ_0; if $I_L = 0$ and $I_R = 1$, then the smoothed rate function is asked to link to the given final rate λ_{k+1}; if $I_L = 1$ and $I_R = 1$, then the smoothed rate function is asked to link to both λ_0 and λ_{k+1}.

In summary, we are given rates λ_i for $i = 1, 2, \ldots, k$ as well as possibly (for the finite-horizon context) an initial rate λ_0 and possibly a final rate λ_{k+1}. In the cyclic case as well as for each of the four finite-horizon cases, we wish to solve for optimal values $\gamma_1^*, \gamma_2^*, \ldots, \gamma_k^*$ of $\gamma_1, \gamma_2, \ldots, \gamma_k$ to obtain the smoothed rate function $\lambda_i - \gamma_i^*, \lambda_i + \gamma_i^*$ for $i = 1, 2, \ldots, k$. As discussed in Section 7, all five minimization problems can be solved exactly and efficiently.

All five cases can arise directly in practice. The cyclic context occurs, for example, when seven rates are given for seven days of the week. The finite-horizon cases arise, for example, when the rate of organ donation is assumed to increase every year. The four finite-horizon cases allow linking to an initial rate λ_0 and/or a final rate λ_{k+1}.

We refer to the process of doubling the number of intervals by minimizing the smoothness objective functions with SMOOTH. When SMOOTH is applied n times iteratively to obtain $2^n k$ intervals, we refer to I-SMOOTH.

7 OPTIMIZING SMOOTHNESS

We now consider the problem of minimizing z_C or z_F. The vector of decision variables is $\underline{\gamma}$; the vector of optimal decision variables is $\underline{\gamma}^* \equiv (\gamma_1^*, \gamma_2^*, \ldots, \gamma_k^*)$. When needed for clarity, we use $\underline{\gamma}_C^*$ and $\underline{\gamma}_F^*$. In both contexts, the smoothed rate function is composed of the $2k$ rates $\tau_{2i-1} = \lambda_i - \gamma_i^*$ and $\tau_{2i} = \lambda_i + \gamma_i^*$ for $i = 1, 2, \ldots, k$.

First consider a trivial special case—the finite-horizon case where $I_L = I_R = 0$ and $k = 1$. Because there are no terms in the summation, $Z_F = 0$, which is not a function of the single decision variable, $(\underline{\gamma}_F)_1$. We declare $(\underline{\gamma}_F^*)_1 = 0$ to be optimal and assume in the rest of this section that we are not discussing this case.

Because the objective functions are quadratic and convex, the unique minimal solutions can be obtained reliably with existing numerical software. Examples include LSQLIN (MathWorks 2010) and HSQP (Betts 1980). Such routines have the advantage that nonnegativity can be integrated into the problem with simple-to-state linear constraints. Our approach, however, is to defer discussion of the nonnegativity constraints to Section 9 so that we can obtain easy-to-compute expressions for $\underline{\gamma}^*$ when the non-negativity constraints are not tight. Our unconstrained expressions for $\underline{\gamma}^*$ require $O(k^2)$ computing effort and $O(k)$ memory requirements.

7.1 Optimal Interval Increments

Results 2 and 3 contain closed-form solutions for $\underline{\gamma}_C^*$ and $\underline{\gamma}_F^*$. We obtained these results by writing the cyclic and finite-horizon quadratic objective functions in matrix form and setting the first derivatives to zero. We solved the resulting system of linear equations to obtain the solutions in terms of a matrix inverse. We simplified the resulting matrix inverses using properties of circulant and symmetric matrices (Gray 1971) for the cyclic context and tri-diagonal matrices (Yamani and Abdelmonem 1997) for the finite-horizon context. The results follow after algebraic simplification.

Result 2 For interval $i = 1, ..., k$ in the cyclic context, the optimal increment change is

$$(\underline{\gamma}_C^*)_i = \frac{2}{(-3)^k - 1} \sum_{h=1}^{\lfloor (k-1)/2 \rfloor} \left[(-3)^{k-h-1} - (-3)^{h-1} \right] (\lambda_{i+h} - \lambda_{i-h}), \qquad (10)$$

where the empty summation for $k = 1, 2$ implies that $(\underline{\gamma}_C^*)_i = 0$.

Result 3 For interval $i = 1,...,k$ in the finite-horizon context, the optimal increment change is

$$(\underline{\gamma}_F^*)_i = \sum_{j=0}^{k+1} w_{i,j}\lambda_j, \tag{11}$$

where the weights $w_{i,j} = \phi_{i,j}/\{4\left[17^{I_L+I_R}(3^{k-1}) - (-1)^{I_L+I_R}(3^{1-k})\right]\}$ for $j = 0,1,\ldots,(k+1)$,

$$\begin{aligned}
\phi_{i,0} &= 12I_L(-1)^i\left[17^{I_R}(3^{k-i}) + (-1)^{I_R}(3^{i-k})\right], \\
\phi_{i,1} &= (-1)^i\{17^{I_L}[1-(3+0.75I_L)I(i\neq 1)] + (-1)^{I_L}[1-(3^{-1}+0.75I_L)I(i\neq 1)]\} \\
&\quad \{17^{I_R}[3^{k-i}-(3^{k-i-1}+0.75I_L3^{k-i})I(i=1)] \\
&\quad +(-1)^{I_R}[3^{i-k}-(3^{i+1-k}+0.75I_L3^{i-k})I(i=1)]\}, \\
\phi_{i,m} &= (-1)^{m-i}(8)\left[17^{I_L}3^{m-2}-(-1)^{I_L}3^{-m}\right]\left[17^{I_R}(3^{k-i})+(-1)^{I_R}(3^{i-k})\right] \\
&\quad \text{for } m = 2,\ldots,(i-1), \\
\phi_{i,i} &= 8\left[17^{I_L}(-1)^{I_R}(3^{2i-k-2}) - 17^{I_R}(-1)^{I_L}(3^{k-2i})\right] \quad \text{if } i\neq 1 \text{ and } i\neq k, \\
\phi_{i,m} &= (-1)^{m-i-1}(8)\left[17^{I_L}(3^{i-1})+(-1)^{I_L}(3^{1-i})\right]\left[17^{I_R}(3^{k-m-1})-(-1)^{I_R}(3^{m-k-1})\right] \\
&\quad \text{for } m = (i+1),\ldots,(k-1), \\
\phi_{i,k} &= (-1)^{k-i-1}\{17^{I_L}[(3^{i-1})-(3^{i-2}+0.75I_R3^{i-1})I(i=k)] \\
&\quad +(-1)^{I_L}[3^{1-i}-(3^{2-i}+0.75I_R3^{1-i})I(i=k)]\}\{17^{I_R}[(3+0.75I_R)I(i\neq k)-1] \\
&\quad +(-1)^{I_R}[(3^{-1}+0.75I_R)I(i\neq k)-1]\}, \\
\phi_{i,k+1} &= 12I_R(-1)^{k-i}\left[17^{I_L}(3^{i-1})+(-1)^{I_L}(3^{1-i})\right],
\end{aligned}$$

and the indicator function $I(A)$ equals 1 if logic A is true and 0, otherwise.

For both the cyclic and finite-horizon contexts, only Results 2 and 3 need to be computed. The results require only $O(k^2)$ computing. Therefore, we avoid numerically minimizing the quadratic objective functions. Better, we avoid computing the $O(k^3)$ terms required for computing the solution using the matrix inverse. Result 3 includes all four finite-horizon cases by writing many terms using the indicators I_L and I_R. Pre-computing a powers-of-three vector speeds computation time in Results 2 and 3.

Equation (10) shows the cyclic behavior of $\underline{\gamma}_C^*$, in which all the λ's with the same lag from γ^*'s have the same weight (i.e., the h-lag weight of λ_{i+h} associated with $(\underline{\gamma}_C^*)_i$ is functionally independent of i). Furthermore, the weight of λ_i for $(\underline{\gamma}_C^*)_i$ is 0.

Equation (11), compared with Equation (10), is more complicated because for each $(\underline{\gamma}_F^*)_i$, the weights of λ's are affected by the end rates and hence, depend on the value of i.

7.2 Asymptotic Approximation

When k is large, we can accurate approximations that are a bit simpler than $\underline{\gamma}_C^*$ and $\underline{\gamma}_F^*$.

Corollary 1 In the cyclic context, when k is large the elements of $\underline{\gamma}_C^*$ can be approximated using

$$(\tilde{\underline{\gamma}}_C^*)_i = \frac{2}{9}\sum_{h=1}^{\lfloor (k-1)/2 \rfloor}(-3)^{1-h}(\lambda_{i+h}-\lambda_{i-h})$$

for $i = 1,2,\ldots,k$.

Corollary 2 In the finite-horizon context, when k is large the elements of $\underline{\gamma}^*_F$ can be approximated using

$$(\tilde{\underline{\gamma}}^*_F)_i = \sum_{j=0}^{k+1} \tilde{w}_{i,j}\lambda_j,$$

where the weights are

$$\begin{aligned}
\tilde{w}_{i,0} &= (I_L/17)(-3)^{2-i}\left[1+(-17)^{-I_R}I(i=k)\right] ,\\
\tilde{w}_{i,1} &= 0.25(-3)^{-i}\left\{3-(1+2.25I_L)I(i=1)+3(-17)^{-I_R}I(i=k)\right\}\\
&\quad \left\{[1-(3+0.75I_L)I(i\neq 1)]+(-17)^{-I_L}\left[1-(3^{-1}+0.75I_L)I(i\neq 1)\right]\right\} ,\\
\tilde{w}_{i,i-h} &= -2(-3)^{-h-1}\Big[1-(-17)^{-I_L}9^{1-i+h}\,I(2\le i\le \lfloor (k+1)/2\rfloor)\\
&\qquad\qquad +(-17)^{-I_R}9^{i-k}\,I(\lfloor (k+1)/2\rfloor+1\le i\le k-1)\Big]\\
&\quad \text{for } h=1,\ldots,(i-2),\\
\tilde{w}_{i,i} &= 2(3^{|2i-k-1|-k})\left[-(-17)^{-I_L}\,I(2\le i\le \lfloor (k+1)/2\rfloor)\right.\\
&\qquad\qquad \left.+(-17)^{-I_R}\,I(\lfloor (k+1)/2\rfloor+1\le i\le k-1)\right] ,\\
\tilde{w}_{i,i+h} &= 2(-3)^{-h-1}\Big[1+(-17)^{-I_L}9^{1-i}\,I(2\le i\le \lfloor (k+1)/2\rfloor)\\
&\qquad\qquad -(-17)^{-I_R}9^{i+h-k}\,I(\lfloor (k+1)/2\rfloor+1\le i\le k-1)\Big]\\
&\quad \text{for } h=1,\ldots,(k-i-1),\\
\tilde{w}_{i,k} &= 0.25(-3)^{i-k-1}[3-(1+2.25I_R)I(i=k)]\left\{[(3+0.75I_R)I(i\neq k)-1]\right.\\
&\quad \left.+(-17)^{-I_R}\left[(3^{-1}+0.75I_R)I(i\neq k)-1\right]\right\} ,\\
\tilde{w}_{i,k+1} &= -(I_R/17)(-3)^{i+1-k}\left[1+(-17)^{-I_L}I(i=1)\right] .
\end{aligned}$$

Although the cyclic large-k approximation $\tilde{\underline{\gamma}}^*_C$ in Corollary 1 is simpler to state than the finite-horizon large-k approximation $\tilde{\underline{\gamma}}^*_F$ in Corollary 2, the values are similar except for the finite-horizon's end effects. The lag-h weight in the cyclic approximation $\tilde{\underline{\gamma}}^*_C$ decreases by the factor $1/3$ each time that $|h|$ increases by 1; the finite-horizon weights differ only a bit, except at intervals near the ends.

With a bit more error, computation effort can be reduced to $O(k)$. Because large-$|h|$ weights are relatively small, weights with lags $|h|$ greater than some value, such as four, can be set to zero. If four is chosen, then the first omitted weight has magnitude about $(2/9)(1/3)^{-4} = 2/729$. The slope-matching heuristic in Section 5 corresponds to truncating the weights after lag $|h| = 1$.

8 EXAMPLES

We now show the results of I-SMOOTH as applied to four examples. Each example is illustrated in a figure, with the first subplots showing the original rate function, and subsequent subplots showing the smoothed rate function after each application of SMOOTH.

Example 1 is given three rates, $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$ in the finite-horizon context, as plotted in the "iteration 0" of Figure 2. The second subplot, "iteration 1", shows six intervals. Notice that the first pair of intervals has rates that average to $\lambda_1 = 1$, the second pair averages to $\lambda_2 = 2$, and the third pair averages to $\lambda_3 = 3$. The "iteration 2" subplot shows twelve intervals, each of length 0.25. Finally, the right-most subplot "iteration 5" shows 96 intervals, each of length $1/32$. This example illustrates a property that we think desirable: the I-SMOOTH limiting rate function should be linear when the original rates have equal jumps.

Example 2, illustrated in Figure 3, is the same as Example 1, except that the context is cyclic rather than finite-horizon. Because the first and third intervals are adjacent, the rates converge as the subplots

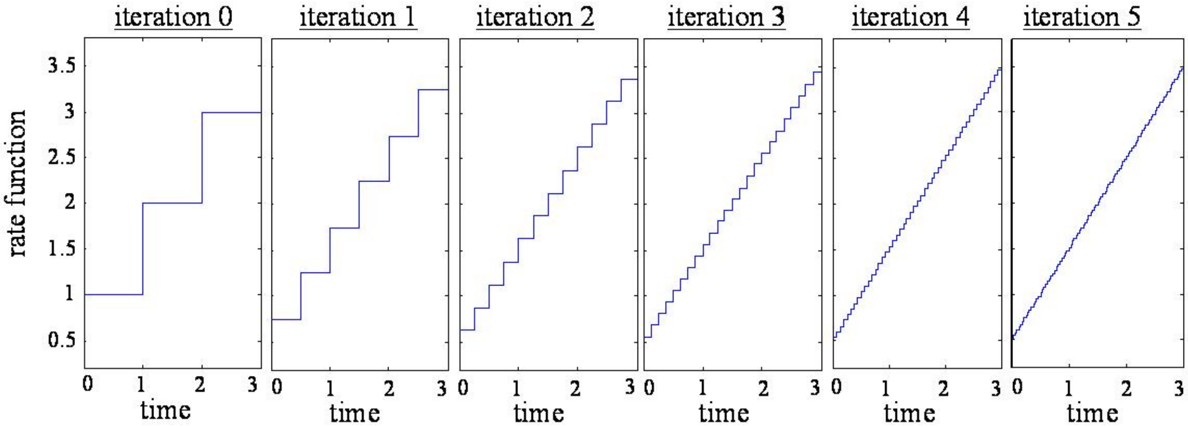

Figure 2: The original rate function and five I-SMOOTH iterations for Example 1. The finite-horizon context with k = 3, $\lambda_1 = 1$, $\lambda_2 = 2$ and $\lambda_3 = 3$.

move from three intervals in iteration 0, to six intervals in iteration 1, through 96 intervals in iteration 5. The limiting I-SMOOTH function is continuous, including at the times $t = 0,3,6,\ldots$ where the cycles connect.

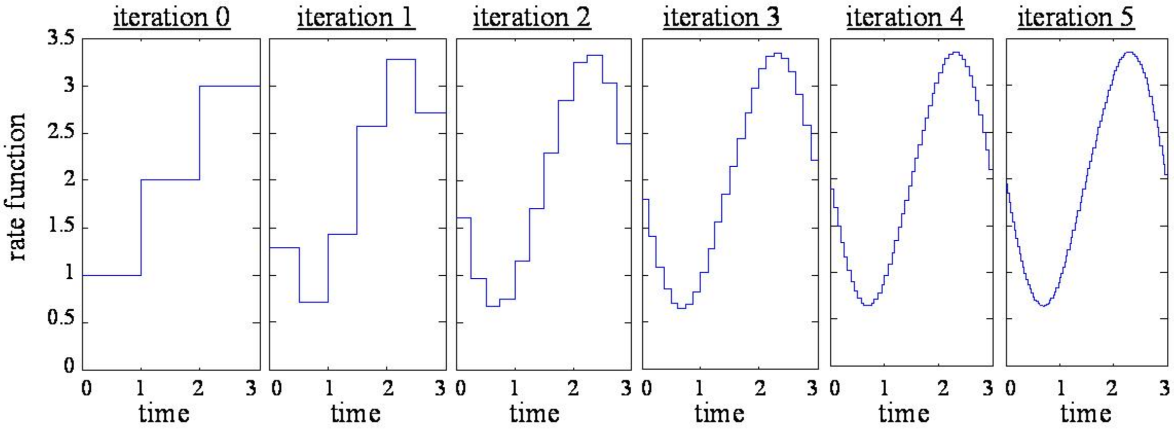

Figure 3: The original rate function and five I-SMOOTH iterations for Example 2. The cyclic context with k = 3, $\lambda_1 = 1$, $\lambda_2 = 2$ and $\lambda_3 = 3$.

Example 3, illustrated in Figure 4, is identical to Example 1, except that the rates are now $\lambda_1 = 0.1$, $\lambda_2 = 1.1$, and $\lambda_3 = 2.1$. Unlike Example 1, the limiting rate function is not linear, despite the constant jumps between interval rates. The reason is that at each iteration the rate function is required to be non-negative and to integrate to the original rate λ_i. The I-SMOOTH limiting function is again continuous, but the limiting rate is zero for the time interval [0, 0.5].

Example 4 is a real-world example. The original piecewise-constant rates, shown in the first subplot of Figure 5, are the 1988 through 2010 yearly live-donor kidney-donation numbers obtained from the Organ Procurement and Transplantation Network web site (OPTN 2011). Because the actual donation-rate function is not piecewise constant, but rather continuous with no jump at the new year, smoothing the original function is reasonable. The second through fourth subplots show the progress of I-SMOOTH iterations 1, 2, and 3, which have 46, 92, and 184 intervals respectively. Non-negativity is not an issue. For

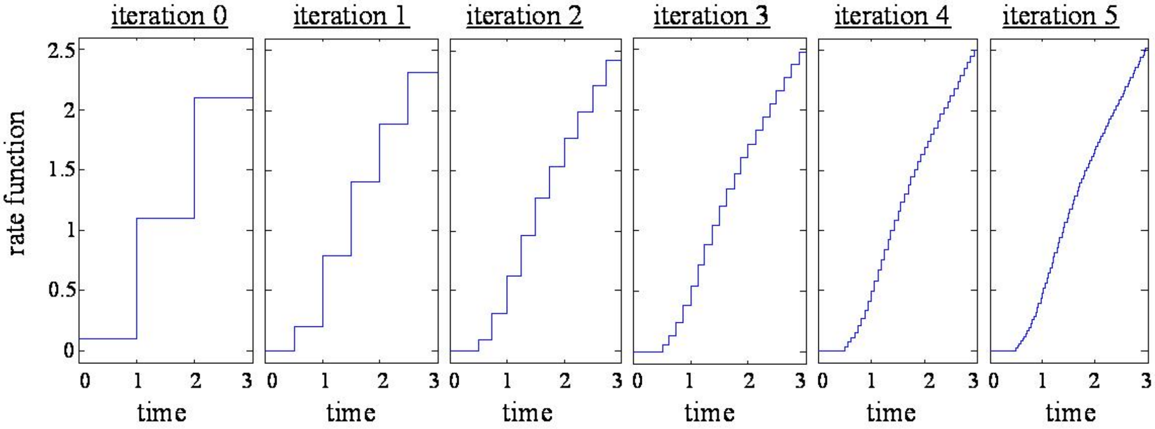

Figure 4: The original rate function and five I-SMOOTH iterations for Example 3. The finite-horizon context with $k = 3$, $\lambda_1 = 0.1$, $\lambda_2 = 1.1$ and $\lambda_3 = 2.1$. Here the limiting result is non-linear because of the non-negativity constraints.

each of the original twenty-three intervals, the expected number of arrivals is unchanged from the original piecewise-constant rates.

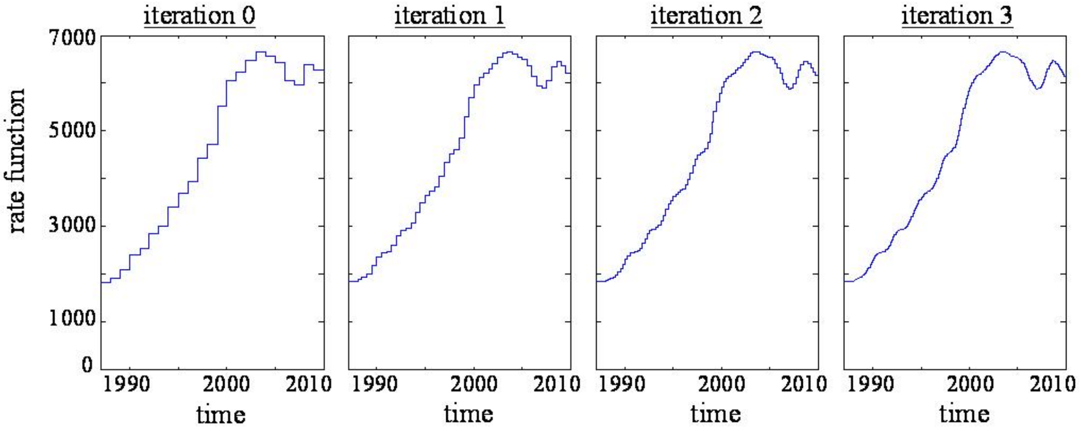

Figure 5: The original rate function and three I-SMOOTH iterations for Example 4. Yearly live-donor kidney-donation data from 1988 through 2010.

9 DISCUSSION

Because of page limitations, we leave five topics undiscussed, other than brief comments here.

Other definitions of smoothness could be used, including the similar concept that minimizes the sum of squared jump sizes. We prefer our definition largely because of a particular example: for the finite-horizon problem, when the given rate constants have equal-sized jumps, our definition yields equal-sized jumps, which are intuitively appealing.

Poisson rate functions must be non-negative. Other than including non-negativity as constraints in numerical optimization, we have considered two ideas. The first is to smooth $\ln(\lambda_i)$, but that approach is complicated by any zero rates. The second is to smooth unconstrained, check for negative rates, “push up” the most negative rate to zero, and re-solve with that rate fixed at zero.

We have no stopping rule for the number of times, say n, to double the number of rates. After doubling the original k intervals n times, the user is left with $2^n k$ piecewise-constant rates. The choice of n is application dependent, but seldom will n be larger than eight.

The computations include powers of 3. The largest power is $3^{2^n k}$. Therefore, iterating many times toward the limiting rate function eventually incurs numerical problems as well as substantial storage and computing. Truncating the weights is an obvious solution to the numerical and computing problems.

After the n iterations, the $2^n k$ rates can be "tipped" in a post-processor to obtain a piecewise-linear rate function. Many heuristics come to mind, such as the slope matching of Section 5. Random-process generation can proceed as in Klein and Roberts (1984).

ACKNOWLEDGMENTS

Our problem formulation and objective function initially appeared in Schmeiser, Rao, and Kumala (2003). Colm O'Cinneide contributed ideas and discussion. This research is supported by Office of Naval Research Contract N000141110419 and Taiwan's National Science Council Grant NSC100-2918-I-033-004.

REFERENCES

Betts, J.T. 1980. Algorithm 559: The Stationary Point of a Quadratic Function Subject to Linear Constraints [E4]. *ACM Transactions on Mathematical Software* 6 (3): 432–436.

Gray, R.M. 1971. Toeplitz and Circulant Matrices: A Review. *Information Systems Laboratory Technical Report, Stanford University.* Available at http://ee.stanford.edu/∼gray/toeplitz.pdf. Published as a paperback by Now Publishers Inc., Boston-Delft, in the series *Foundations and Trends in Communications and Information Theory* 2(3): 155–329, 2005.

Klein, R. W. and Roberts, S. D. 1984. A time-varying Poisson arrival process generator. *Simulation* 42: 193–195.

Kuhl, M.E. and Wilson, J.R. 2000. Least squares estimation of nonhomogeneous Poisson processes. *Journal of Statistical Computation and Simulation* 67: 75–108.

Kuhl, M.E. and Wilson, J.R. 2001. Modeling and simulating Poisson processes having trends or nontrigonometric cyclic effects. *European Journal of Operational Research* 133 (3): 566–582.

Leemis, L.M. 1991. Nonparametric estimation of the cumulative intensity function for a nonhomogeneous Poisson process. *Management Science* 37 (7): 886–900.

MathWorks 2010. R2010b MathWorks Documentation. Available at http://www.mathworks.com/help/toolbox/optim/ug/lsqlin.html .

OPTN 2011. Organ Procurement and Transplantation Network, http://optn.transplant.hrsa.gov/latestData/advancedData.asp [accessed April 2011].

Schmeiser B., Rao R., and Kumala, N. 2003. Smoothing piecewise-constant rate functions. *Proceedings of the Industrial Engineering Research Conference*, Institute of Industrial Engineers, 6 pages (cd only).

Yamani, H.A. and Abdelmonem, M.S. 1997. The analytic inversion of any finite symmetric tridiagonal matrix. *Journal of Physics A: Mathematical and General* 30 (8): 2889–2893.

AUTHOR BIOGRAPHIES

HUIFEN CHEN is professor of Industrial and Systems Engineering Department at Chung-Yuan University, Taiwan. She completed her Ph.D. in Industrial Engineering at Purdue University in 1994 and master in statistics at Purdue University in 1990. Her research interests include statistical process control, public health, and stochastic root finding. Her email address is huifen@cycu.edu.tw.

BRUCE SCHMEISER has been active within the Winter Simulation Conference for many years. His e-mail address is bruceschmeiser@gmail.com and his web page is http://gilbreth.ecn.purdue.edu/∼bruce/.