Problem 1 We wish to simulate from a Poisson process on $[0, t_0]$ whose rate function $\dot{\mu}(t)$ is bounded above by a finite M, using a source of uniform variables $U_1, U_2, \ldots \overset{\text{iid}}{\sim} U(0,1)$. Below U denotes a new $U(0,1)$ variable each time it appears. We use the following algorithm:

1. first generate $N \sim \text{Poiss}(Mt_0)$. Suppose that $N = n$;
2. then generate $U_1, \ldots, U_n \overset{\text{iid}}{\sim} U(0,1)$ and set $T_1 = t_0 U_1, \ldots, T_n = t_0 U_n$;
3. then generate $U_1^*, \ldots, U_n^* \overset{\text{iid}}{\sim} U(0,1)$ and retain T_j only if $MU_j^* \le \dot{\mu}(T_j)$;
4. return the retained values of $T_1, \ldots, T_n$.

(a) Show that if $U \sim U(0,1)$, $a \in \mathbb{R}$ and $b > 0$, then $a + bU \sim U(a, a+b)$. Hence give the distributions of the T_j and of the MU_j^*.

(b) At the rejection step 3, show that the probability that T_j is retained is $\int_0^{t_0} \dot{\mu}(t)\,\mathrm{d}t/(Mt_0) = \mu(t_0)/(Mt_0)$, and deduce that the probability that $T_j = t$, conditional on it being retained, is $\dot{\mu}(t)/\mu(t_0)$. Use the independence of the T_j to explain why the algorithm achieves its purpose.

(c) The efficiency of such an algorithm can be defined as the ratio of the expected number of T_js output to the expected number of Us used. Show that this equals $\mu(t_0)/(2Mt_0)$, and deduce that it is optimal to take $M = \sup_{0 \le t \le t_0} \dot{\mu}(t)$. Can you think of a way to improve on this algorithm?

Problem 2 The events $t_1, \ldots, t_n$ of a Poisson process on $(0, t_0]$ are available, and it is supposed that the intensity function is of the form $\dot{\mu}(t) = \exp\{\sum_{r=1}^{p} \beta_r b_r(t)\}$, where the functions $b_r(t)$ are basis functions defined on $[0, t_0]$ (e.g., polynomials, $b_r(t) = t^{r-1}$).

(a) Show that the corresponding log likelihood can be written in the form

$$\ell(\beta) = \sum_{r=1}^{p} \beta_r s_r - k(\beta), \quad \beta = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p,$$

and give formulae for s_r and $k(\beta)$. Do you recognise this? What are the implications?

(b) The calculation of $k(\beta)$ may be painful. Suppose instead that $[0, t_0]$ is divided into K disjoint intervals of lengths $\Delta = t_0/K$, and let $y_1, \ldots, y_K$ be the numbers of events in the successive intervals. Explain why approximate inference on β can be based on the log likelihood

$$\ell_K(\beta) = \sum_{k=1}^{K} (y_k \log \mu_k - \mu_k),$$

where $\mu_k = \Delta \dot{\mu}\{(k - 1/2)\Delta\} = \Delta \exp\{\sum_{r=1}^{p} \beta_r b_r((k - 1/2)\Delta)\}$. In what sense is this approximate? Is this model also an exponential family?

(c) If $\dot{\mu}(t)$ is bounded and continuous, show that $\ell_K(\beta) - n \log \Delta \to \ell(\beta)$ as $K \to \infty$.

Problem 3 This question uses the ideas from the previous one to fit the model with $\dot{\mu}(t) = \lambda e^{\beta t}$ to the Bengal data. In a *generalized linear model (GLM)* the mean μ of a response variable y with an exponential family distribution (normal, binomial, Poisson, gamma, ...) can depend nonlinearly on a *linear predictor* $x^{\mathrm{T}}\beta$, where x is a vector of known covariates and β is to be estimated. The usual GLM for a Poisson response sets $y \sim \text{Poiss}(\mu)$ and $\log \mu = o + x^{\mathrm{T}}\beta$, with o a known term called an *offset*. The following code fits this model with $\log \mu(t) = \log \Delta + \log \lambda + \beta t$, where $\log \Delta$ is the offset, and $K = 20$ intervals. It uses the histogram function `hist` to obtain the counts in the K intervals and the function `glm` to fit the Poisson model:

```
load("bengal.dat")
K <- 20; t0 <- 101; Delta <- t0/K
breaks <- c(0:K)*Delta
(y <- hist(bengal-1877,breaks=breaks,plot=FALSE)$counts)
t <- Delta*(c(0:(K-1))+0.5)
log.Delta <- rep(1,K)*log(Delta)
summary(glm(y~1+t+offset(log.Delta),family=poisson))
```

Part of the output looks like

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.110184   0.188686  -0.584  0.55925   #  log lambda from slide 37
t            0.008224   0.002942   2.795  0.00518 **  # beta from slide 37

Null deviance: 42.550  on 19  degrees of freedom
Residual deviance: 34.601  on 18  degrees of freedom  # difference is 42.55-34.60=7.95
```

(a) Compare the output above with the results on slide 38. Do they agree adequately, in your opinion?

(b) Try increasing K, and see at what point the results stabilise. Discuss.

(c) If you have nothing better to do, try fitting some other more complex models, e.g., fitting periodic functions using

```
c <- cos(2*pi*t)
s <- sin(2*pi*t)
summary(glm(y~t+offset(log.Delta)+s+c,family=poisson))
```

and using the residual deviances with and without `s+c` to test whether the added terms are needed.