Biometrika (2012), **99**, 1, *pp*. 1–9

Printed in Great Britain

Advance Access publication on 31 July 2012

Supplemental Material For: Bayesian Local Extremum Splines

BY M. W. WHEELER

National Institute for Occupational Safety and Health
1150 Tusculum Avenue
Cincinnati, OH 45226, MS C-15
mwheeler@cdc.gov

D. B. DUNSON AND A. H. HERRING

Department of Statistical Science
Duke University
Box 90251, Durham, NC 27708
dunson@duke.edu amy.herring@duke.edu.

1. INSERTION OF A SPLINE BETWEEN MODELS

Let $\mathcal{M}$ and $\mathcal{M}'$ be two models that differ by one knot; model $\mathcal{M}'$ has one extra knot that is a child of a node also in $\mathcal{M}$. If the knot set for model $\mathcal{M}$ is $\{0{\cdot}0000, 0{\cdot}0625, 0{\cdot}1250, 0{\cdot}2500, 0{\cdot}5000, 0{\cdot}7500, 1{\cdot}0000\}$ and the knot set for model $\mathcal{M}'$ is $\{0{\cdot}0000, 0{\cdot}0625, 0{\cdot}1250, 0{\cdot}2500, 0{\cdot}5000, 0{\cdot}6250, 0{\cdot}7500, 1{\cdot}0000\}$, then these two models differ only by the knot $0{\cdot}6250$. If $j = 1$ and $\mathcal{M}$ and $\mathcal{M}'$ are defined as above, there are 8 local extremum splines in model $\mathcal{M}'$. Of the 8 local extremum splines, 5 of these are shared with $\mathcal{M}$. This relationship is shown pictorially in Fig. 1. Model $\mathcal{M}'$ is plotted in red and model $\mathcal{M}$ is plotted in black, and there are only three basis functions in $\mathcal{M}'$ that do not have an identical basis functions in $\mathcal{M}$.

The sharing of basis functions between nested models allows for the development of the reversible jump algorithm outlined in the manuscript. The number of functions shared depends upon the order of the B-spline used in the construction. For example, when $j = 1$, $p(\mathcal{M} \mid Y, \beta_{-\mathcal{M}})$ is a mixture distribution with 4 components and $p(\mathcal{M}' \mid Y, \beta_{-\mathcal{M}'})$ is a mixture distribution with 8 components. If $j = 2$, there are 8 and 16 components in the mixture distribution. Computation time significantly increases with j, because the mixture distribution has 2^j components for model $\mathcal{M}'$ and 2^{j-1} components for model $\mathcal{M}$.

2. DEGREE OF THE B-SPLINE USED

The construction of the local extremum spline is dependent upon the degree j of the B-spline. In other applications of B-splines, $j = 3$ is a standard choice, leading to cubic B-splines. For local extremum splines, we find it is sufficient practically to set $j = 1$; this may produce slightly less smooth function estimates than $j = 2$ or $j = 3$, but we find differences are very minor. Table 1 compares integrated mean square error in estimating functions f_1, f_2, f_3, f_4, f_5, f_6, and f_7 in the manuscript, when $n = 200$. There are noticible differences only for function f_5, and these minor differences, illustrated in Fig. 2, are due to $j = 1$ producing a slightly less smooth estimate.

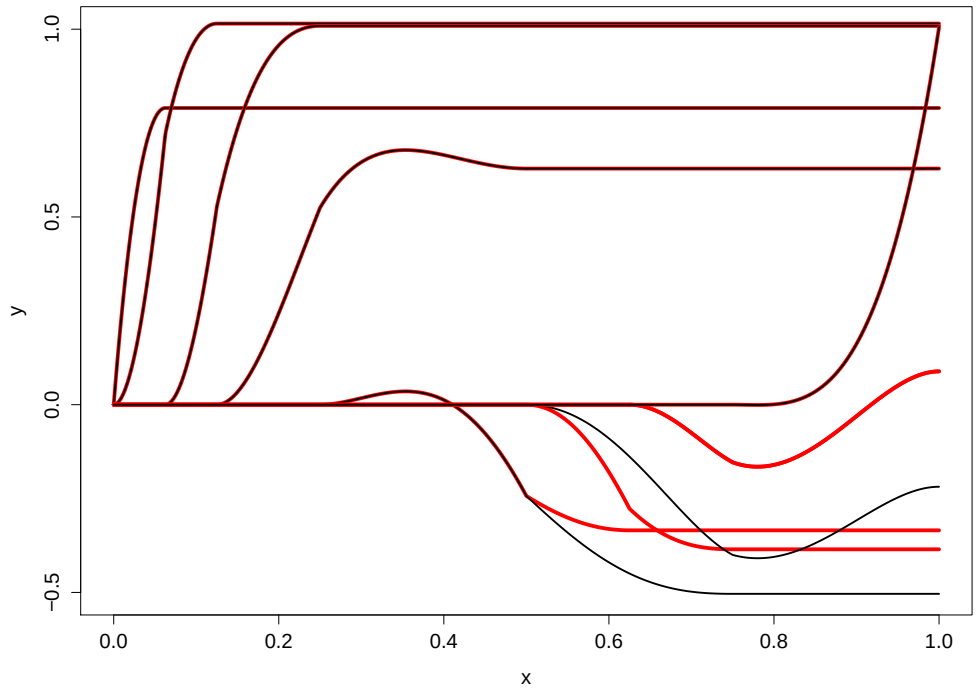

Fig. 1. Plot of two B-spline bases, red and black, that differ by only one knot.

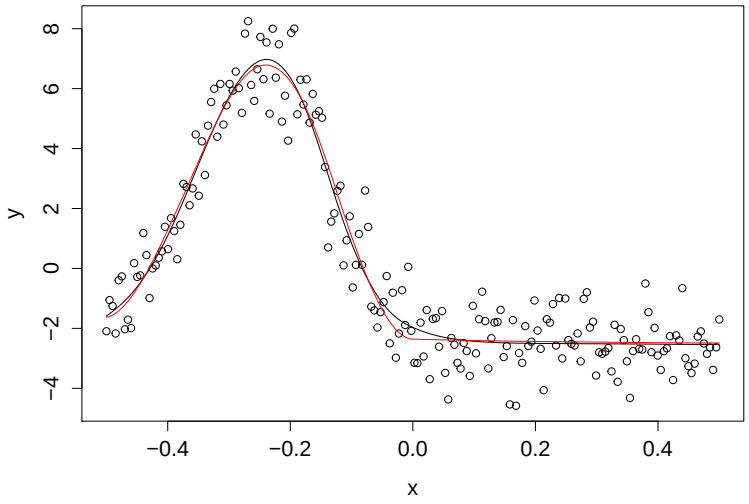

Fig. 2. The fit of the local extremum spline with $j = 2$, black line, compared to a local extrema spline with $j = 1$, red line.

Table 1. *Integrated mean squared error comparing the local extremum spline constructions when $j = 1$ and $j = 2$. For each function, the top row represents the simulation condition $\sigma^2 = 4$ and the bottom row represents the simulation condition $\sigma^2 = 1$.*

True Function	Spline Construction $j = 1$	Spline Construction $j = 2$
f_1	0·096	0·091
	0·022	0·024
f_2	0·104	0·148
	0·040	0·044
f_3	0·081	0·088
	0·024	0·024
f_4	0·088	0·091
	0·026	0·025
f_5	0·200	0·116
	0·055	0·032
f_6	0·115	0·123
	0·032	0·034
f_7	0·120	0·123
	0·030	0·032

Computational concerns are important in choosing j. As j increases, it is more computationally demanding to compute $p(\mathcal{M} \mid Y, \beta_{-\mathcal{M}})$ and $p(\mathcal{M}' \mid Y, \beta_{-\mathcal{M}})$. When $j = 1$ the Markov chain Monte Carlo algorithm takes between 20 and 50 seconds per $50,000$ iterations, and when $j = 2$ the algorithm takes between 50 and 90 seconds per $50,000$ iterations.

3. Improving Sampling Through Parallel Tempering

The posterior distribution is often multimodal, and the sampler proposed in the manuscript often gets stuck in local modes. This occurs when widely different parameter values have relatively large support by the data, and there is low posterior density between these isolated modes. To increase the probability of jumps between modes, a parallel tempering algorithm (Geyer, 1991, 2011) is implemented. Define m parallel chains over $h_i(y, \theta_i) = \exp\{\kappa_i \ell(y \mid \theta_i) + \log p(\theta_i)\}$, where $\theta_i = \{\mathcal{M}_i, \beta_i, \alpha_i, \pi_i, \lambda_i, \sigma_i\}$, $\ell(y \mid \theta_i)$ is the log-likelihood of the data given θ_i, $p(\theta_i)$ is the prior over the parameters, and $0 < \kappa_1 < \ldots < \kappa_m = 1$.

The sampling algorithm proceeds by first running chains for each h_i independently. Then, for two adjacent chains i and j chosen with equal probability, where chains are defined adjacent when $j = i + 1$, the parameters θ_i and θ_j are swapped in a Metropolis-Hasting step. The acceptance probability is $\min\{1, r(i, j)\}$ with

$$r(i, j) = \frac{h_j(y, \theta_i) h_i(y, \theta_j)}{h_i(y, \theta_i) h_j(y, \theta_j)}.$$

With a good choice for $\kappa_1 < \ldots < \kappa_m = 1$, mixing of the target distribution $h_m(y, \theta_m)$ is improved, and accurate posterior estimates of the function, as well as the number of change points in the model, can be obtained with relatively few iterations.

Table 2. *Effective sample sizes per* $50,000$ *samples for seven functions given in Section 4.2 of the manuscript.*

True function	Effective Sample Size							
	$g(0)$	$g(0{\cdot}2)$	$g(0{\cdot}4)$	$g(0{\cdot}6)$	$g(0{\cdot}8)$	$g(1)$	α_1	α_2
$g_1(x)$	457	1013	988	542	1749	1425	460	411
$g_2(x)$	376	1456	721	794	1301	1860	501	569
$g_3(x)$	678	1229	1209	1543	1736	1254	540	567
$g_4(x)$	490	1700	1794	1743	2006	1987	580	596
$g_5(x)$	742	1256	1225	1123	1934	1663	702	696
$g_6(x)$	699	1169	1629	1561	1885	1939	707	593
$g_7(x)$	769	873	1881	1940	1661	1423	709	685

4. CONVERGENCE OF THE MARKOV CHAIN MONTE CARLO ALGORITHM

To determine the number of samples needed in the simulation studies, simulated data sets were fit and convergence was monitored. In all examples, the local extremum spline is defined as in the manuscript with $H = 2$ and $j = 2$. Table 2 shows the effective sample size per $50,000$ Markov chain Carlo samples for the simulated functions in section 4.2 of the manuscript for the function at $\{0, 0{\cdot}2, 0{\cdot}4, 0{\cdot}6, 0{\cdot}8, 1{\cdot}0\}$ and the change point parameters α_1 and α_2, which are computed using CODA (Plummer et al., 2006). This table shows that the effective sample size for each parameter is typically over $1,000$ per $50,000$ samples when estimating the function; the effective sample size for the change point parameters is over $1,000$ per $150,000$ samples.

We also monitored the mixing of the algorithm. Figures 3 and 4 show the observed trace plots for the change point parameters. Figure 3 shows how the change point parameters, which are not individually identifiable, move between extremum. In this example, the function $4{\cdot}5\sin\{2\pi(x - 0{\cdot}5)\}$ is sampled evenly across the interval $[-0{\cdot}5, 0{\cdot}5]$. A total of $n = 200$ points are taken and the error distribution is $N(0, 1)$. This function has well defined change points at $-0{\cdot}25$ and $0{\cdot}25$. The posterior distribution reflects this by placing a high probability on two change points concentrated at these locations.

Fig. 4 shows a trace plot for $40,000$ samples from a posterior distribution with less distinct change points. Here, 200 points are sampled from the function $-20(x - 0{\cdot}25)^2$ with $\epsilon_i \sim N(0, 4)$. To model an umbrella shaped pattern, one change point must be interior to $[-0{\cdot}5, 0{\cdot}5]$ and the other change point must be less than or equal to $-0{\cdot}5$. The extrema is not well identified and the change points alternate between an umbrella shape and a monotone increasing pattern. The plot shows the parameters moving between these shapes. This curve estimates are shown in Fig. 5. The local extremum approach, black line, is compared against a frequentist smoothing approach estimated using the R (R Core Team, 2015) function smooth.spline(), red line. Both compared to the true curve, green line. This plot shows the difficulty both methods have in determining the single maximum given the data. The local extremum spline forms a flat line from the maximum to the right hand side of the interval, while the smoothing spline produces artifactual bumps in this region.

5. PROTECTING THE TYPE I ERROR RATE.

We investigate the type I error rate using specified cut points for hypothesis tests defined in proposition 2 for the simulations in the manuscript. For this comparison, we use the cut point

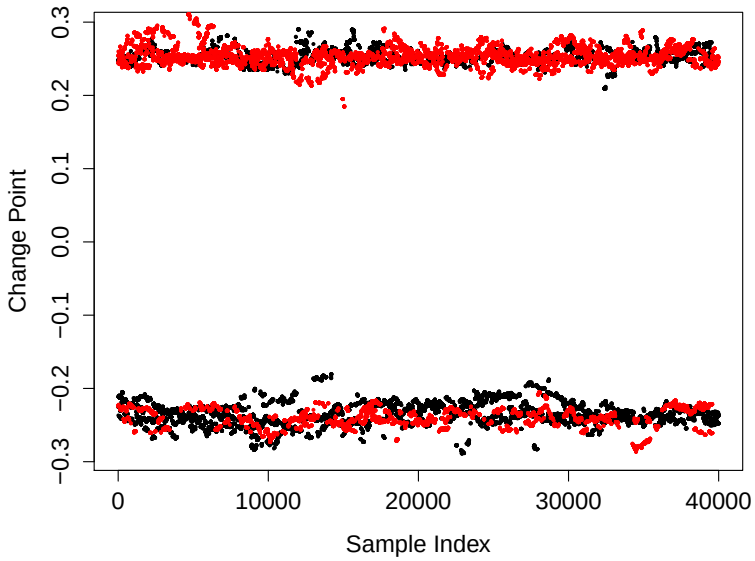

Fig. 3. The trace plot of 40,000 sampled change point parameters when $H = 2$ in the LX-spline. Black dots represent one change point and red dots represent the other. The sampled function has two change points.

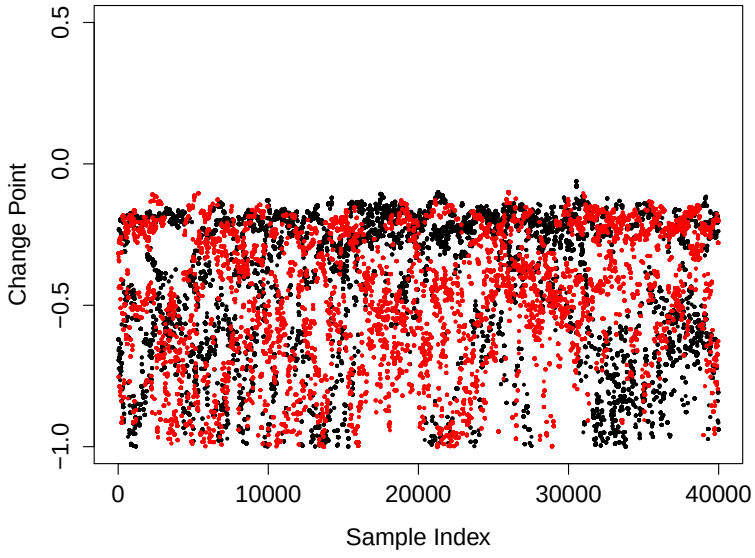

Fig. 4. The trace plot of 40,000 sampled change point parameters when $H = 2$ in the LX-spline. Black dots represent one change point and red dots represent the other. The sampled function has two change points.

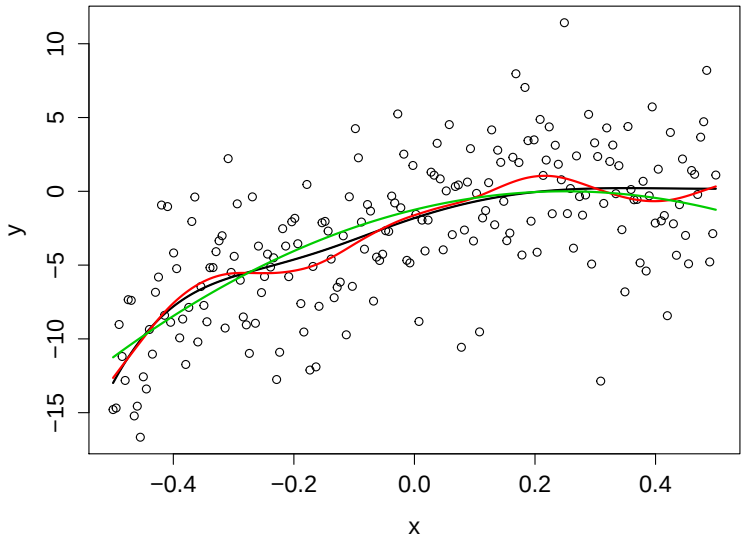

Fig. 5. The fit of the local extremum spline, black line, compared to frequentist smoothing splines, red line, when estimating the true curve, green line.

$10^{1/2}$ suggested by Jeffreys (1998, page 432) and a cut point that is based upon the distribution of the Bayes factor, under the null hypothesis, in a worst case scenario (Baraud et al., 2005).

To define the worst case scenario, 100 evenly spaced points were chosen in the interval $[0, 1]$ with the response at each point simulated from a $N(0, 1)$. This is a flat curve that is the boundary for any test defined in proposition 2. For each simulated data set, a local extremum spline with $H = 2$ is fit and the Bayes factor is computed. This is done $1,000$ times for each hypothesis, and the distribution of the Bayes factor under the null is estimated. For this simulation, the cut point is chosen such that $\alpha = 0{\cdot}05$. All hypothesis testing simulations described in the manuscript are used when the true curve is: monotone increasing, denoted as hypothesis $\mathbb{H}_{01}$, and any shape having 0 or 1 extremum in the interval, denoted as hypothesis $\mathbb{H}_{02}$. Simulation function f_5 was also considered using the same conditions in the hypothesis testing section of the manuscript.

Table 3 gives the observed percentage of times the null was correctly chosen over the alternative. This table shows that using the cut points of $3{\cdot}16$, Jeffreys, or $3{\cdot}47$ and $3{\cdot}75$, Baraud et al. (2005) for $\mathbb{H}_{01}$ and $\mathbb{H}_{02}$ respectively, the null is correctly chosen at a rate greater than the specified 95% rate for all but one case, and all cases are within the margin of error $0{\cdot}085$. This suggests that the type I error is protected at $\alpha = 0{\cdot}05$ using standard cut points for Bayes factors.

6. SAMPLING ALGORITHM OF PARAMETERS GIVEN $\mathcal{M}$

In outlining the algorithm, we drop the dependence on $\mathcal{M}$ from the design matrix and decompose the design matrix $B^*(\alpha)$ as

$$B^*(\alpha) = z_r(\alpha)B^{*r} + \ldots z_0(\alpha)B^{*0}.$$

Table 3. *The observed* $100(1-\alpha)\%$ *of times where the listed hypothesis was correctly chosen.*

Function	$\mathbb{H}_{01}$		$\mathbb{H}_{02}$	
	Jeffreys	Baraud et al. (2005)	Jeffreys	Baraud et al. (2005)
g_1	100	100	99·6	99·6
g_3	100	100	100	100
g_4	N/A	100	100	100
g_2	100	N/A	92·4	99·6
f_5	N/A	N/A	100	100

Here B^r, $0 \leq r \leq H$, is a $n \times k$ matrix where each element $B^{*r}_{(i,k)}$ from row i and column k of the matrix B^{*r} is computed as

$$B^{*r}_{(i,k)} = \int_{\tau_k}^{x_i} \xi^r B_{(j,k)}(\xi)\, d\xi.$$

Define $z_r(\alpha)$ as a function of α corresponding to the coefficient of x^r in the polynomial $\prod_{i=1}^{H}(x-\alpha_h)$. For example, when $H=2$ one has $z_0(\alpha)=\alpha_2\alpha_1$, $z_1(\alpha)=-(\alpha_1+\alpha_2)$ and $z_2(\alpha)=1$. The matrix $B^*(\alpha)$ is used when sampling β and $\{B^{*r}\}_{r=0}^{H}$ is used when sampling α.

Sampling Algorithm

1. For $1 \leq k \leq K+j-1$, when sampling β_k, let $Y^* = Y - B^*(\alpha)_{-k}\beta_{-k}$. Where β_{-k} is β without entry k, and $B(\alpha)_{-k}$ is the design matrix without column k. Letting $w = B^*(\alpha)_k$, a $n \times 1$ column vector representing column k in $X(\alpha)$, sample β_k from

$$p(\beta_k|\mathcal{M}) \propto 1_{(\beta_k=0)}\frac{\phi(0,\hat{E},\hat{V})}{\lambda} + 1_{(\beta_k>0)}\phi(\beta_k,\hat{E},\hat{V}),$$

   where $\hat{V} = \{\tau(w'w)\}^{-1}$, $\hat{E} = \hat{V}\,(\tau w'Y^* - \lambda)^{-1}$.
2. Let $Y^* = Y - B^*(\alpha)_{-0}\beta_{-0}$ and sample $\beta_0 \sim N(E,V)$ where $V = (\tau n + c^{-1})^{-1}$ and $E = V(\tau Y^*)$.
3. For each α_h in α, define $Y^* = Y - [\sum_{r=0}^{H}\{z_r^-(\alpha,\alpha_h)B^{*r}\}]\boldsymbol{\beta}$, where $z_r^-(\alpha,\alpha_h)$ is a function representing the terms in $z_r(\alpha)$ that do not have α_h as a coefficient. For example, when $H=2$ then $\prod_{i=1}^{H}(x-\alpha_h) = x^2 - (\alpha_1+\alpha_2)x + \alpha_1\alpha_2$; in this case, $z_2(\alpha)=1$, $z_2^-(\alpha,\alpha_1)=1$, $z_1(\alpha)=-(\alpha_1+\alpha_2)$, $z_1^-(\alpha,\alpha_1)=-\alpha_2$, and $z_0^-(\alpha,\alpha_1)=0$. Similarly, let $w = [\sum_{r=0}^{H}\{z_r^*(\alpha,\alpha_h)B^{*r}\}]\boldsymbol{\beta}$, where $z_r^*(\alpha,\alpha_h)$ is a function that contains only the terms in $z_r(\alpha)$ with α_h factored out. Again, when $H=2$ for α_1, $z_0^*(\alpha,\alpha_1)=\alpha_2$, $z_1^*(\alpha,\alpha_1)=-1$, and $z_2^*(\alpha,\alpha_1)=0$ as no term in $z_2^*(\alpha)$ contains α_1. Given these quantities sample

$$\alpha_h \propto N(E,V)1_{a\leq\alpha_h\leq b}$$

   where $V = \{\tau(w'w) + D_h^{-1}\}^{-1}$, $E = V(\tau w'Y^* + D_h^{-1}C_h)$, and $C_h = (b-a)/2$, $D_h = 1$.
4. Sample $\lambda \propto Ga\left\{\sum_{k=1}^{K+j-1}1_{(\beta_k=0)} + \delta, \kappa + \sum_{k=1}^{K+j-1}\beta_k 1_{(\beta_k>0)}\right\}1(\lambda > 1e-5)$, a truncated gamma distribution.
5. Sample $\pi \sim Beta\left\{\nu + \sum_{k=1}^{K+j-1}1_{(\beta_k=0)}, \omega + K + j - 1 - \sum_{k=1}^{K+j-1}1_{(\beta_k=0)}\right\}$

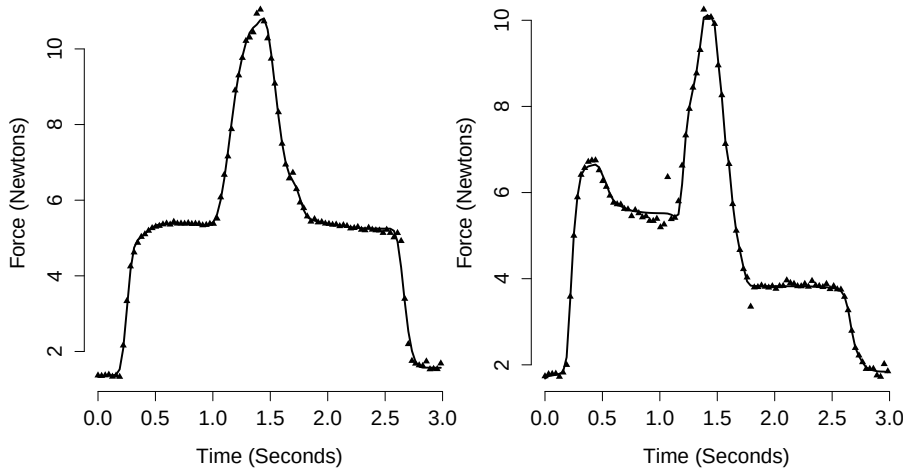

Fig. 6. Fit of the local extremum spline, black line, to observed muscle force data, solid triangles.

7. APPLICATIONS: ESTIMATING MUSCLE FORCE

When studying the ability of a muscle to adapt to exercise protocols, muscle force tracings are often used. One approach involves first activating the muscle and then after a short period of time moving the joint through the range of motion (Baker et al., 2008). It is expected that the muscle force quickly obtains a maximum force with the observed force decreasing until joint movement; however, the observed force may plateau and not decrease before movement. When the joint is moved, there is an expected increase in the force output until the joint reaches a specific angle, after which, the observed force decreases until the joint reaches its original position. When the joint returns to its original position, the muscle remains activated and the force output is non-increasing until deactivation. Estimation of this muscle force curve may allow better understanding of adaptation or maladaptation following exercise, but it is important to include known biophysical constraints in curve estimation.

We model two force tracings, with $n = 96$ per tracing, using a local extremum spline having at most $H = 3$ local extrema. Consistent with prior knowledge of a very high signal to noise ratio, we place a $\text{Ga}(2000, 1)$prior on σ^{-2}. We also applied frequentist smoothing splines, Gaussian processes, and Bayesian P-splines. Competing methods are close to interpolating the data points, leaving unwanted artifactual bumps in the function estimate. However, as seen in Fig. 6, the local extremum spline obtains an estimate restricted to the known shape and robust to minor local fluctuations. Further, when the force tracing exhibits a single maxima, as in the left plot, the local extremum spline can readily distinguish between this shape, and a shape which has two maximum, as in the right plot, with no change in the model.

REFERENCES

BAKER, B. A., HOLLANDER, M. S., MERCER, R. R., KASHON, M. L. & CUTLIP, R. G. (2008). Adaptive stretch-shortening contractions: diminished regenerative capacity with aging. *Applied Physiology, Nutrition, and Metabolism* **33**, 1181–1191.

BARAUD, Y., HUET, S. & LAURENT, B. (2005). Testing convex hypotheses on the mean of a Gaussian vector. application to testing qualitative hypotheses on a regression function. *Annals of Statistics* **33**, 214–257.

GEYER, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, E. M. Keramidas, ed. Red Hook, NY: Interface Foundation of North America.

GEYER, C. J. (2011). Importance sampling, simulated tempering and umbrella sampling. In *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones & X. Meng, eds. Boca Raton, FL: Chapman & Hall/CRC, pp. 295–311.

JEFFREYS, H. (1998). *The Theory of Probability*. OUP Oxford.

PLUMMER, M., BEST, N., COWLES, K. & VINES, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News* **6**, 7–11.

R CORE TEAM (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[*Received April* 2012. *Revised September* 2012]