Implementation of 'Wp' smoothing for EoR foreground fitting

Geraint Harker

November 27, 2008

1 Introduction

At a given point on the sky, LOFAR measures a brightness temperature $T(\nu)$ as a function of frequency, ν. We assume that we have a datacube in which this function $T(\nu)$ has three components: the cosmological signal from neutral hydrogen during the epoch of reionization, astrophysical foregrounds and noise. The foregrounds are much brighter than the cosmological signal, but are postulated to be smooth as a function of frequency. In this article we describe an algorithm to estimate the contribution from the foregrounds as a function of frequency by fitting $T(\nu)$ with a smooth curve. After subtracting this fit, in an ideal case the residuals contain contributions only from the noise and the cosmological signal.

Other methods of fitting the foregrounds have various drawbacks. Fitting a given functional form — a power law or a polynomial of some specified order, say — can introduce a bias if this form is incapable of accurately representing the foregrounds. A form with many parameters can also be vulnerable to 'over-fitting', where the signal we are interested in gets fitted out because the fitting is too sensitive to small-scale features coming from the cosmological signal and noise. This also appears to be a serious problem with most 'non-parametric' fitting methods, such as smoothing splines. In addition, because these methods penalize curvature (rather than *change* in curvature) they are subject to the well-known problem of 'attrition' when fitting a curved function.

We use, instead, the 'Wp' smoothing method described by Mächler (1993, 1995). 'Wp' stands for 'Wendepunkt', the German word for 'inflection point'. This method penalizes changes in the curvature of the fitting function. We briefly review this method in Section 2 to give some background and to establish some notation. Unfortunately, while this method gives us a system of differential equations to solve which can be written down concisely, the computation of the solution presents some difficulties. We have implemented a method to solve them, which we describe in Section 3.

2 Overview of the method

We have a set of observations $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ which we wish to fit with a smooth function $f(x)$. Each y_i may have an associated error, σ_i. The inflection points of f are, by definition, the zeros of f''. We assume the inflection points are known and label them w_i, $i = 1, 2, \ldots, n_w$. Then we may write

$$f''(x) = p_{\mathbf{w}}(x)\mathrm{e}^{h_f(x)} \tag{1}$$

where

$$p_{\mathbf{w}}(x) \stackrel{\text{def}}{=} s_f(x - w_1)(x - w_2)\ldots(x - w_{n_w}) \quad , \tag{2}$$

$s_f = \pm 1$ and h_f is a function as many times differentiable as f''.

We may then express the problem as follows. We wish to find the function f which minimizes

$$\sum_{i=1}^{n} \rho_i(y_i - f(x_i)) + \lambda \int_{x_1}^{x_n} h_f'(t)^2 \mathrm{d}t \tag{3}$$

where λ is a Lagrange multiplier, the integral term measures the change in curvature 'apart from inflection points' and the function ρ_i determines the size of the penalty incurred when $f(x_i)$ deviates from y_i. For simple least-squares minimization, for example, $\rho_i(\delta) = \frac{1}{2}\delta^2 \ \forall i$.

The solution of this minimization problem must then satisfy the following ordinary differential equation (ODE):

$$h_f'' = p_{\mathbf{w}}\mathrm{e}^{h_f} L_f \tag{4}$$

where, using the notation $a_+ = \max(0, a)$,

$$L_f(x) = -\frac{1}{2\lambda}\sum_{i=1}^{n}(x - x_i)_+ \psi_i(y_i - f(x_i)) \tag{5}$$

and $\psi_i(\delta) = \frac{\mathrm{d}}{\mathrm{d}\delta}\rho_i(\delta)$. The solution must satisfy the 'multi-boundary' conditions

$$h_f'(x_1) = h_f'(x_n) = \sum_i \psi_i(y_i - f(x_i)) = \sum_i x_i\psi_i(y_i - f(x_i)) = 0 \quad . \tag{6}$$

We may write ψ_i explicitly as $\psi_i(\delta) = \delta$ for least squares, or, taking the errors into account, as $\psi_i(\delta) = \delta/\sigma_i$. Alternatively, a more robust method may use

$$\psi_i(\delta) = \begin{cases} c & \delta/\sigma_i > c \\ \delta/\sigma_i & |\delta/\sigma_i| \le c \\ -c & \delta/\sigma_i < -c \end{cases} \tag{7}$$

for some $c > 0$.

Not only are the boundary conditions problematic, but the ODE itself, Eqn. 4, includes on the right-hand side a contribution from $f(x_i)$ for each x_i, meaning that the equation is not in the 'standard form' assumed by off-the-shelf solvers for boundary value problems (BVPs).

Note, also, that the minimization is performed with s_f and $\{w_i\}$ fixed. To apply the procedure to an arbitrary data set, then, requires a further minimization over the number and position of the inflection points. We therefore require some method to give a starting approximation for f, f', h_f, h'_f, n_w, $\{w_i\}$ and s_f.

In principle we should also like some method to choose the Lagrange multiplier, λ. This has no 'natural' value, and indeed the method remains well defined for $\lambda \to 0$ and $\lambda \to \infty$. Mächler suggests using the autocorrelation function of the residuals. This could be problematic for our application, since there may be real correlations in the noise between frequency bands due to the cosmological signal, and in any case it still requires some level of arbitrary choice. In practice, we simply choose a reasonable-looking value for λ.

3 Implementation

3.1 BVP solver

The first task is to rewrite Eqn. 4 as a system of coupled first-order equations. This is simply done as follows (recalling that $p_{\mathbf{w}}$ is a function of x only, since the parameters s_f and $\{w_i\}$ are fixed), where we drop the f subscript on h_f and write out the function L_f explicitly to make the dependence on f clear:

$$h'(x) = g(x) \quad ; \tag{8}$$

$$g'(x) = p_{\mathbf{w}}(x)\mathrm{e}^{h(x)}\left[-\frac{1}{2\lambda}\sum_{i=1}^{n}(x-x_i)_+\psi_i(y_i-f(x_i))\right] \quad ; \tag{9}$$

$$f'(x) = k(x) \quad ; \tag{10}$$

$$k'(x) = p_{\mathbf{w}}(x)\mathrm{e}^{h(x)} \quad . \tag{11}$$

Eqns. 8 and 10 define our new functions g and k respectively. We may then trivially reexpress the boundary conditions as

$$g(x_1) = 0 \quad ; \tag{12}$$

$$g(x_n) = 0 \quad ; \tag{13}$$

$$\sum_i \psi_i(y_i - f(x_i)) = 0 \quad ; \tag{14}$$

$$\sum_i x_i\psi_i(y_i - f(x_i)) = 0 \quad . \tag{15}$$

Eqns. 8–15 are not in the standard form required by most BVP solvers. A typical solver requires the user to provide a function $\mathbf{\Phi}$ defined by $\mathbf{y}' = \mathbf{\Phi}(x, \mathbf{y}(x))$ where each element of $\mathbf{\Phi}$ corresponds to one of the equations in the system. It also requires a function $\mathbf{\Gamma}$ such that the equation $\mathbf{\Gamma}(\mathbf{y}(a), \mathbf{y}(b)) = 0$ expresses the boundary conditions, with a and b being the endpoints of the interval on which the problem is specified. Unfortunately, to evaluate Eqn. 9 requires us to know $f(x_i)$ for $i = 1, \ldots, n$ as well as $f(x)$, while Eqns. 14 and 15, defining two of the the boundary conditions, involve the points $x_2, \ldots, x_{n-1}$ as well as x_1 and x_n.

In fact, dealing with the unusual boundary conditions is not such a severe problem. Ascher & Russell (1981) catalogue a variety of ways to put a system of differential equations into standard form, and an elegant method is available for integral constraints such as Eqns. 14 and 15.

First consider Eqn. 14. We rewrite the left-hand side as the integral of a piecewise-constant function (a step function) on the interval $[x_1, x_n]$. That is, it becomes

$$\int_{x_1}^{x_n} G(t, f(t))\mathrm{d}t = 0 \tag{16}$$

where

$$G(t, f(t)) = \begin{cases} \frac{2\psi_1(y_1 - f(x_1))}{x_2 - x_1} & x_1 \le t < \frac{x_1+x_2}{2} \\ \frac{2\psi_m(y_m - f(x_m))}{x_{m+1} - x_{m-1}} & \frac{x_{m-1}+x_m}{2} \le t < \frac{x_m+x_{m+1}}{2} \qquad 1 < m < n \\ \frac{2\psi_n(y_n - f(x_n))}{x_n - x_{n-1}} & \frac{x_{n-1}+x_n}{2} \le t \le x_n \end{cases} \tag{17}$$

Now let

$$V(x) = \int_{x_1}^{x} G(t, f(t))\mathrm{d}t \quad . \tag{18}$$

Then $V'(x) = G(x, f(x))$, $V(x_1) = 0$ and $V(x_n) = 0$. In other words, we have rewritten the boundary condition (14) by adding the function $V(x)$ to our system of equations, with V' defined as shown and satisfying the boundary conditions $V(x_1) = V(x_n) = 0$. $V'(x)$ is easy to compute because it is piecewise-constant, and the new boundary condition is enforced at the endpoints of the interval and does not require the value of any functions at interior points.

Similarly, we may rewrite Eqn. 15 by introducing a new function $W(x)$, satisfying $W(x_1) = W(x_n) = 0$ and with a derivative $W'(x) = H(t, f(t))$

given by

$$H(t,f(t)) = \begin{cases} \frac{2x_1\psi_1(y_1-f(x_1))}{x_2-x_1} & x_1 \le t < \frac{x_1+x_2}{2} \\ \frac{2x_m\psi_m(y_m-f(x_m))}{x_{m+1}-x_{m-1}} & \frac{x_{m-1}+x_m}{2} \le t < \frac{x_m+x_{m+1}}{2} \qquad 1 < m < n \\ \frac{2x_n\psi_n(y_n-f(x_n))}{x_n-x_{n-1}} & \frac{x_{n-1}+x_n}{2} \le t \le x_n \end{cases} \tag{19}$$

Thus we can see that the boundary conditions do not, in themselves, present a special problem. The equations for $V'(x)$ and $W'(x)$ do, however, suffer from the same problem as Eqn. 9; that is, they require knowledge of $f(x_i)$ in addition to $f(x)$. This appears to be a more difficult issue than the one of boundary conditions which are not evaluated at the endpoints. We therefore take an alternative approach, again suggested by Ascher & Russell (1981).

We split the domain of solution into $n-1$ intervals, $[x_1, x_2]$, $[x_2, x_3]$, $\ldots$, $[x_{n-1}, x_n]$. In each interval we change variables, letting

$$t = \frac{x - x_m}{x_{m+1} - x_m} \quad \text{for} \quad x_m \le x \le x_{m+1} \tag{20}$$

which maps each interval onto the unit interval, $[0, 1]$. Then, on this interval, we define functions $f_m(t)$, $g_m(t)$, $h_m(t)$, $k_m(t)$, $p_{\mathbf{w},m}(t)$ for $m = 1, 2, \ldots, n-1$ such that, for $x_m \le x \le x_{m+1}$, $f_m(t) = f(x)$, $g_m(t) = g(x)$, $h_m(t) = h(x)$, $k_m(t) = k(x)$ and $p_{\mathbf{w},m}(t) = p_{\mathbf{w}}(x)$. We further define the functions $q_m(t)$ for $m = 1, \ldots, n$ where $q_m(t) = f_m(0)$ for $m = 1, \ldots, n-1$ and $q_n(t) = f_{n-1}(1)$. Our system of four equations (8–11) then becomes the following system of $5n - 4$ equations (where dashes now indicate differentiation with respect to t, and we have used the chain rule where necessary):

$$\begin{aligned} f'_m(t) &= (x_{m+1} - x_m)k_m(t) && (21)\\ k'_m(t) &= (x_{m+1} - x_m)p_{\mathbf{w},m}(t)\mathrm{e}^{h_m(t)} && (22)\\ h'_m(t) &= (x_{m+1} - x_m)g_m(t) && (23)\\ g'_m(t) &= (x_{m+1} - x_m)p_{\mathbf{w},m}(t)\mathrm{e}^{h_m(t)} \\ &\quad \times \left\{ \frac{-1}{2\lambda} \sum_{i=1}^{m} [x_m + (x_{m+1} - x_m)t]\psi_i(y_i - q_i(t)) \right\} && (24)\\ q'_j(t) &= 0 && (25) \end{aligned}$$

where the index m runs from 1 to $n-1$ and j runs from 1 to n. The q functions carry the value of f at the data points, $f(x_i)$, to the interior of the intervals, a property which is imposed with the boundary conditions

$$\begin{aligned} q_m(0) &= f_m(0) \quad \text{for} \quad m = 1, \ldots, n-1; && (26)\\ q_n(0) &= f_{n-1}(1) \quad . && (27) \end{aligned}$$

Our original boundary conditions become

$$\begin{aligned} g_1(0) &= 0 \quad ; & (28)\\ g_{n-1}(1) &= 0 \quad ; & (29)\\ \sum_{i=1}^{n} \psi_i(y_i - q_i(0)) &= 0 \quad ; & (30)\\ \sum_{i=1}^{n} x_i \psi_i(y_i - q_i(0)) &= 0 \quad . & (31) \end{aligned}$$

The remaining $4(n-2)$ boundary conditions come from imposing continuity on the functions $f(x)$, $g(x)$, $h(x)$ and $k(x)$:

$$\begin{aligned} f_m(1) &= f_{m+1}(0) \quad ; & (32)\\ g_m(1) &= g_{m+1}(0) \quad ; & (33)\\ h_m(1) &= h_{m+1}(0) \quad ; & (34)\\ k_m(1) &= k_{m+1}(0) \quad ; & (35) \end{aligned}$$

where here the index m runs from 1 to $n-2$.

Note that the boundary conditions only involve the value of functions at $t = 0$ and $t = 1$, and that to calculate the derivatives given by Eqns. 21–25 at a given value of t only requires the evaluation of functions at the same value of t. The system is therefore suitable for solution using the MATLAB routine `bvp4c`, which we call with an initial mesh of five evenly spaced points.

3.2 Finite difference scheme

Our implementation using a standard BVP solver appears, at present, to be slow and somewhat unstable. This could be because the solver does not exploit the special form of the problem. We have therefore tried a different approach: discretizing the differential equation into a finite difference equation defined on some grid, and then solving the resulting nonlinear algebraic system. This approach is similar to standard relaxation methods, though the form of the problem again means that some tricks used for speeding up relaxation methods do not apply. The main drawback we anticipate, though, is that the size of the grid used to discretize the system must be chosen beforehand, and is not adaptive.

As before, we have data points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ and our fitting function has inflection points at $w_1, w_2, \ldots, w_{n_w}$. We choose a mesh such that the abscissae of the data points are also mesh points. That is, we have a mesh $X_1, X_2, \ldots, X_N$, where $N \geq n$, and where $X_{m_i} = x_i$ for $i = 1, \ldots n$, with $m_1 = 1$ and $m_n = N$. That is, m_i gives the position in the mesh of the i^{th} data point, and the first and last data points define the ends of the grid.

Let $f(X_i) = f_i$ and $h(X_i) = h_i$ (which implies that $f(x_i) = f_{m_i}$). Then we may discretize Eqn. 1 as

$$\frac{f_{j+1} - 2f_j + f_{j-1}}{(X_{j+1} - X_j)(X_j - X_{j-1})} = p_{\mathbf{w}}(X_j)\mathrm{e}^{h_j} \quad \text{for} \quad j = 2, \ldots, N-1. \tag{36}$$

Defining $\Delta_j = (X_{j+1} - X_j)(X_j - X_{j-1})$, we have

$$f_{j+1} - 2f_j + f_{j-1} - \Delta_j p_{\mathbf{w}}(X_j)\mathrm{e}^{h_j} = 0 \quad . \tag{37}$$

Similarly, we may rewrite Eqn. 4 as

$$h_{j+1} - 2h_j + h_{j-1} - \Delta_j p_{\mathbf{w}}(X_j)\mathrm{e}^{h_j} \left[-\frac{1}{2\lambda} \sum_{i=1}^{n} (X_j - x_i)_+ \psi_i(y_i - f_{m_i}) \right] = 0 \tag{38}$$

where in each case the index j runs from 2 to $N-1$. Since the grid and the inflection points do not change, $\Delta_j p_{\mathbf{w}}(X_j)$ can be precomputed for efficiency.

The boundary conditions now become:

$$h_2 - h_1 = 0 \quad ; \tag{39}$$

$$h_N - h_{N-1} = 0 \quad ; \tag{40}$$

$$\sum_i \psi_i(y_i - f_{m_i}) = 0 \quad ; \tag{41}$$

$$\sum_i x_i \psi_i(y_i - f_{m_i}) = 0 \quad . \tag{42}$$

We therefore have $2N$ algebraic equations for the $2N$ unknowns, $f_1, \ldots, f_N$ and $h_1, \ldots, h_N$. Note that it is unnecessary to express the system in terms of first-order equations, and indeed this would be undesirable as it would increase the size of the system. We solve the system of equations, (37)–(42), using the MATLAB routine `fsolve`. This provides a selection of optimization algorithms to solve the general nonlinear problem $\mathbf{F}(\mathbf{Y}) = \mathbf{0}$; we have found that the 'Trust-Region-Reflective' algorithm works best. As is the case for the BVP solver, speed is improved by providing an analytical Jacobian matrix

$$\frac{\partial F_\mu}{\partial Y_\nu} \quad \text{for} \quad \mu, \nu = 1, \ldots, 2N \tag{43}$$

where $Y_i = f_i$ and $Y_{N+i} = h_i$ for $i = 1, \ldots, N$, and $\mathbf{F} = \mathbf{0}$ expresses Eqns. 37–42. Calculating the Jacobian is fiddly but the procedure should be clear, and so is not reproduced here.

3.3 Providing an initial guess

It is generally important to specify a reasonable initial guess for the solution of a boundary value problem. In many cases, the system has two or more

valid solutions and the right starting point is required to pick out the desired one. Otherwise, a good guess may be needed to ensure that the solver converges quickly (or indeed at all) on the solution.

Mächler recommends the use of log-splines to provide initial estimates for s_f, $\{w_i\}$ and f. Our implementation is at present less general. For EoR foreground fitting, we assume that the foregrounds have no inflection points (as would be true for a superposition of power laws with negative indices) and fit a power law to obtain an initial estimate for s_f and f.

References

Ascher, U. & Russell, R. D. 1981, SIAM Review, 23, 238

Mächler, M. 1993, Very smooth nonparametric curve estimation by penalizing change of curvature

—. 1995, Annals of Statistics, 23, 1496