Journal of Sound and Vibration (1999) **222**(3), 512–520
Article No. jsvi.1998.2072, available online at http://www.idealibrary.com on IDEAL

COMPUTATION OF THE HARMONICS-TO-NOISE RATIO OF A VOICE SIGNAL USING A FUNCTIONAL DATA ANALYSIS ALGORITHM

J. C. LUCERO
Departamento de Matemática, Universidade de Brasília, DF 70910-900, *Brazil*

(*Received* 23 *November* 1998)

1. INTRODUCTION

The harmonics-to-noise ratio (HNR) has been used to quantify the cycle-to-cycle waveform irregularity of voice signals [l]. It assumes that the signal consists of two components: a harmonic component which is the periodic pattern that repeats from cycle-to-cycle, and an additive noise component which produces the cycle-to-cycle differences. In the cited work [1], the former was computed as the average of the individual cycles (wavelets), and the latter as the difference of the wavelets to their average. Since the wavelets have different lengths, they were normalized in time by zero padding each wavelet to the longest period, so that they could be compared on a point-by-point basis.

The zero-padding normalization has a disadvantage [2]: since the wavelets differ in length, a large portion of the computed HNR will be caused by the length variance. A first solution to this problem would be linear expansion or compression of all wavelets to a common length. However, phase differences between landmarks of the wavelets (maxima, minima, inflection points, etc.) would still remain, which would contaminate the computed HNR. Therefore, an accurate computation of the HNR requires a non-linear expansion or compression of the wavelets (non-linear time normalization) to align their landmarks in time. To accomplish an optimal wavelet alignment, dynamic programming algorithms have been used [2–4]. As an alternative approach, zero phase transformations have been used to remove all phase-related information from the wavelets prior to computation of the HNR [3]. However, this approach had in general poorer results than the dynamic programming algorithms.

A similar issue has been recently discussed in the case of speech movement signals [5], where various approaches for extracting the average from a set of speech wavelets were compared. Special attention was given to non-linearly normalized averaging, in which the wavelets are non-linearly expanded to a common length and optimally aligned before extracting the average, and a new algorithm for optimal alignment based on Functional Data Analysis (FDA) [6–8] was introduced. It was argued that this algorithm might have advantages over previous dynamic programming because the resultant normalizing functions are smooth and differentiable, it does not require users to select one of the wavelets as reference (template), and different optimization criteria may be adopted according to the application.

0022–460X/99/180512 + 09 $30.00/0

This letter will show the application of the FDA algorithm to compute the HNR, of a voice signal. Its general purpose is to show the potential of FDA to signal analysis, and stimulate further developments and applications. The algorithm will be first described in detail and its application will be next illustrated with an example.

2. FDA ALGORITHM

2.1. NON-LINEAR TIME NORMALIZATION

Let us call the set of wavelets to normalize (i.e., the individual cycles of a voice signal) $x_i(t)$, where $i = 1, \ldots, N$, and N is the number of wavelets. For simplicity, they are considered as continuous functions of time. Further, they are assumed to have the same length, from $t = 0$ to $t = 1$.

For each wavelet $x_i(t)$, one wants to determine a strictly increasing and reasonably smooth transformation of time $h_i(t)$, (warping function), such that each normalized wavelet

$$x_i^*(t) = x_i[h_i(t)] \tag{1}$$

is close in some measure to its average

$$\bar{x}^*(t) = \frac{1}{N}\sum_{i=1}^{N} x_i^*(t) \tag{2}$$

Such a transformation may be conveniently defined as

$$h_i(t) = A_0 + A_1 \int_0^t \mathrm{e}^{\int_0^u w_i(v)\,\mathrm{d}v}\,\mathrm{d}u, \tag{3}$$

where $w_i(t)$ is the relative curvature of $h_i(t)$ (to be determined optimally) and coefficients A_0 and A_1 are selected so that $h_i(0) = 0$ and $h_i(1) = 1$ [6–8]. Given any function $w_i(t)$ such that the integrals in equation (3) exist, this equation will produce a strictly increasing and twice differentiable function $h_i(t)$.

Different measures may be used to evaluate the closeness of the normalized records to their average, according to the particular application [7]. Here, the measure

$$F(x_i,w_i,\alpha_1,\alpha_2,\lambda) = \alpha_1 \int_0^1 [\bar{x}^*(t) - x_i^*(t)]^2\,\mathrm{d}t + \alpha_2 \int_0^1 \left[\frac{\mathrm{d}\bar{x}^*}{\mathrm{d}t}(t) - \frac{\mathrm{d}x_i^*}{\mathrm{d}t}(t)\right]^2 \mathrm{d}t$$
$$+ \lambda \int_0^1 w_t^2(t)\,\mathrm{d}t \tag{4}$$

is adopted, where α_1, α_2, and λ are positive, constants. The first integral is the classical squared error measure used in the dynamic programming algorithms [2–4]. The second integral incorporates the first derivative into the measure, to achieve a better alignment of maxima, minima, and inflection. points of the waveforms [9]. The third integral incorporates a penalty for the roughness of the warping function, controlled by parameter λ (the larger the value of λ, the smaller the curvature of h_i).

Hence, the problem consists of estimating the curvature functions $w_i(t)$ in equation (3) that will minimize the total measure (cost function)

$$C(x_1, \ldots, x_N, w_1, \ldots w_N, \alpha_1, \alpha_2, \lambda) = \sum_{i=1}^{N} F(x_i, w_i, \alpha_1, \alpha_2, \lambda). \tag{5}$$

The curvature functions $w_i(t)$ may be defined by subdividing the time interval [0, 1] with a set of $K+1$ equally spaced breakpoints τ_k, with $k = 0, \ldots, K$ and $0 = \tau_0 < \tau_1 \cdots < \tau_K = 1$, and setting

$$\int_0^t w_i(v)\,\mathrm{d}v = \sum_{k=1}^{K} c_k \Phi_k(t) \tag{6}$$

where c_k are the parameters to determine optimally, and Φ_k is the hat function

$$\Phi_k(t) = \begin{cases} (t - \tau_{k-1})/\Delta & \text{if } t \in [\tau_{k-1}, \tau_k] \\ (\tau_{k+1} - t)/\Delta & \text{if } t \in [\tau_k, \tau_{k+1}], \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\Delta = 1/K$ is the separation between consecutive breakpoints, and $\Phi_K(t)$ is not defined in interval $[\tau_K, \tau_{K+1}]$ [6]. With this simple definition, equation (3) may be integrated analytically, with the result

$$h_i(t) = h_i(\tau_{k-1}) + A_1 \frac{\Delta \mathrm{e}^{c_{k-1}}}{c_k - c_{k-1}} [\mathrm{e}^{(c_k - c_{k-1})(t-\tau_{k-1})/\Delta} - 1], \quad \text{for } t \in [\tau_{k-1}, \tau_k]. \tag{8}$$

Note that when computing equation (4), the average $\bar{x}^*$ of the normalized wavelets x_i^* would be required before knowing the whole set of these normalized wavelets. The following iterative process is then used. Coefficients c_k are given initial values such as $c_k^{(0)} = 0$ ($k = 1, \ldots, K$), which yields the initial approximations for the warping functions $h_i^{(0)}(t) = t$ and normalized wavelets $x_i^{*(0)}(t) = x_i(t)$. The average $\bar{x}^{*(0)}(t)$ of this initial set of normalized wavelets is computed with equation (2), and used in equation (4) to determine a new set of coefficients $c_k^{(1)}$. These values of $c_k^{(1)}$ are then used to obtain a better estimate of the warping functions $h_i^{(1)}(t)$ and the normalized wavelets $x_i^{*(1)}(t) = x_i[h_i^{(1)}(t)]$. A new average $\bar{x}^{*(1)}(t)$ is next determined, and the process iterated obtaining a new set of normalized wavelets at each iteration, until there is no significant difference between two consecutive iterations. For example, the iterations may be stopped when the difference between the total cost from two consecutive iterations $|C^{(n)} - C^{(n-1)}|$ is smaller than a precision parameter ε.

When applying this algorithm to discretions, the integrals in equation (4) are replaced by summations.

The algorithm assumes that all the wavelets had the same length from 0 to 1. It is possible to modify the above equations to accommodate wavelets of different lengths and time spans. However. it is much simpler to interpolate all wavelets to a common length and attribute to this length an artificial [0, 1] time span, before applying the non-linear normalization (typically, this interpolation will be done after extracting all F_0 perturbation measures). The final results should be the same in both cases.

2.2. COMPUTATION OF THE HNR

After the wavelets have been optimally aligned in time using the above algorithm, the HNR may be computed. The HNR is defined as [1]

$$\mathrm{HNR} = \frac{N \int_0^1 \bar{x}^{*2}(t)\,\mathrm{d}t}{\sum_{i=1}^{N} \int_0^1 [\bar{x}^*(t) - x_i^*(t)]^2\,\mathrm{d}t}. \tag{9}$$

The denominator would be equal to the total optimization measure, if only the first integral in equation (4) is used (i.e., the classical squared error criterion). In that case, the non-linear normalization would try to align the wavelets so as to achieve the maximum value of HNR. The present algorithm will also align the wavelets and increase in general the value of the HNR, but within the constraints imposed by the requirements of smoothness and alignment of derivatives, as controlled by parameters α_1, α_2 and λ.

3. EXAMPLE

The acoustical signal of a subject producing a sustained /*e*/ at comfortable pitch and intensity level for more than 1 s was recorded on a WAV format file at a sampling frequency of 44 100 Hz and 16-bit resolution. After the recording, all further processing was done using Matlab software. The signal was edited and a stable segment of 20 cycles was visually selected. The boundaries of individual cycles were identified using the method of zero crossings, with the aid of low-pass filtering [10]. Figure 1 shows the extracted 20 cycles (wavelets). Their lengths vary from 359 samples (8·14 ms) to 369 samples (8·37 ms).

To apply the time normalization algorithm, a value of $K = 5$ (number of time breaks minus one) was first selected. All the wavelets were then interpolated to 371 samples (the Matlab implementation of the algorithm requires the number of samples minus 1 to be a multiple of K) using cubic splines, and an artificial time span from 0 to 1 was adopted for the interpolated wavelets. The time normalization algorithm was applied using the parameters $\alpha_1 = 10^{-2}$, $\alpha_2 = 2 \times 10^{-6}$, and $\lambda = 10^{-1}$ in equation (4). The values of α_1 and α_2 were selected so as to obtain values of the integrals in equation (4) in the units range. Working in this range of values facilitated the application of the Matlab functions. An adequate value of λ was selected by visual inspection of the results.

Although the noise in the wavelets is small, some degree of smoothing was necessary to obtain a good estimate of the derivatives. These derivatives were therefore computed using a fourth order and 16-point Savitsky–Golay algorithm [11]. Figure 2 shows a typical wavelet and its first derivative.

The minimization of equation (5) was done using a Broyden–Fletcher–Goldfarb–Shanno quasi-Newton algorithm [11]. The computation of successive

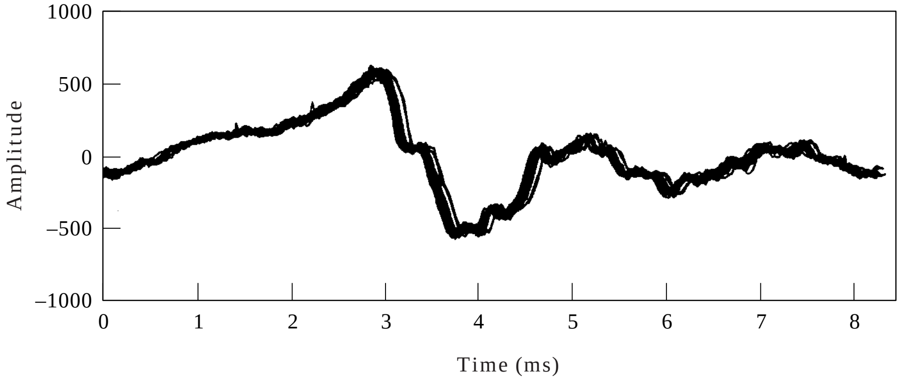

Figure 1. Un-normalized voice wavelets.

approximations to the optimal warping functions was iterated until the difference between the total cost measure in two consecutive iterations was less than $\varepsilon = 0{\cdot}1$. Figure 3 shows the resulting time normalized wavelets.

Figure 4 shows the optimal warping functions $h_i(t)$ (top). Since the amount of warping required is small, the warping functions are very close to the straight line $h_i^{(0)}(t) = t$. The warping may be better illustrated by computing the difference $h_i(t) - t$, which represents the departure of normalized time from

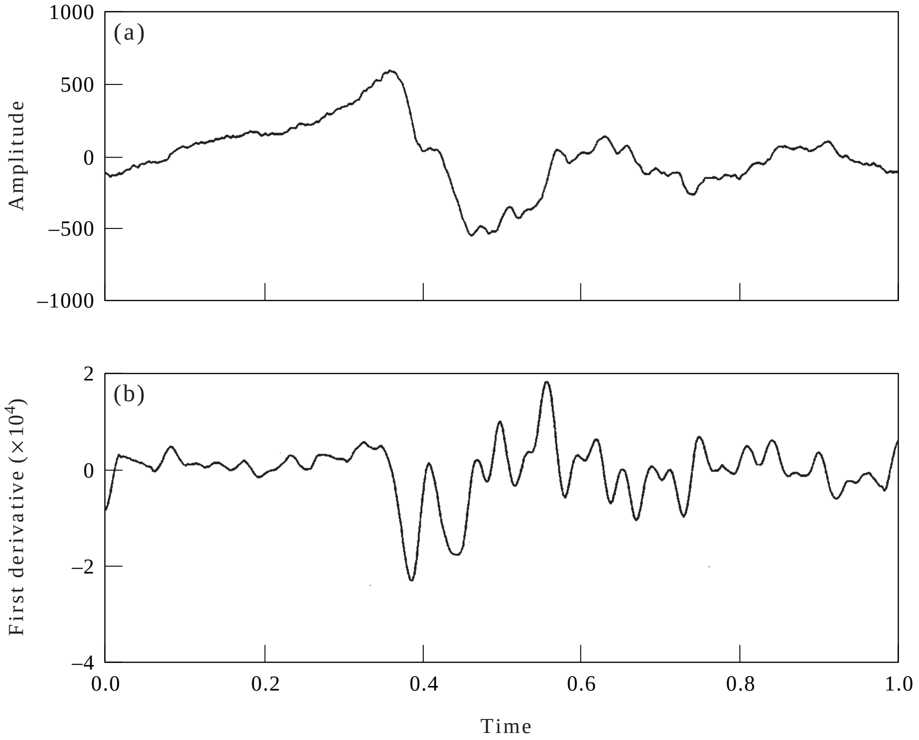

Figure 2. (a) Typical voice wavelet and (b) its first derivative.

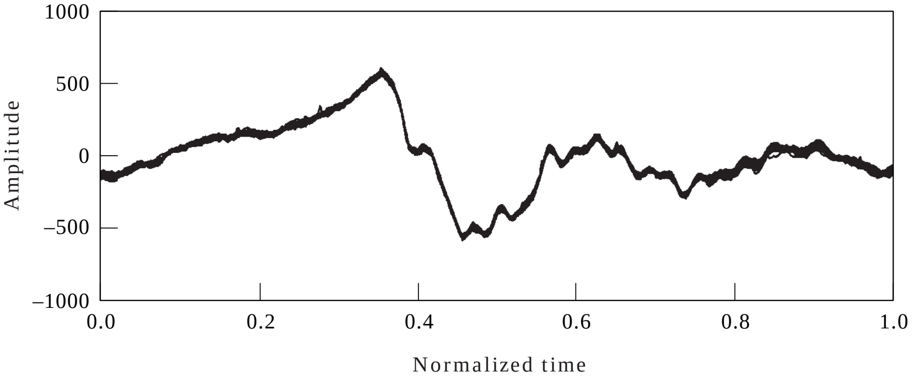

Figure 3. Non-linearly normalized voice wavelets.

original time (Figure 4, bottom). We may define some index to measure the amount of warping required, such as the mean rms value

$$W = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\int_0^1 [h_i(t) - t]^2 \, \mathrm{d}t}. \tag{10}$$

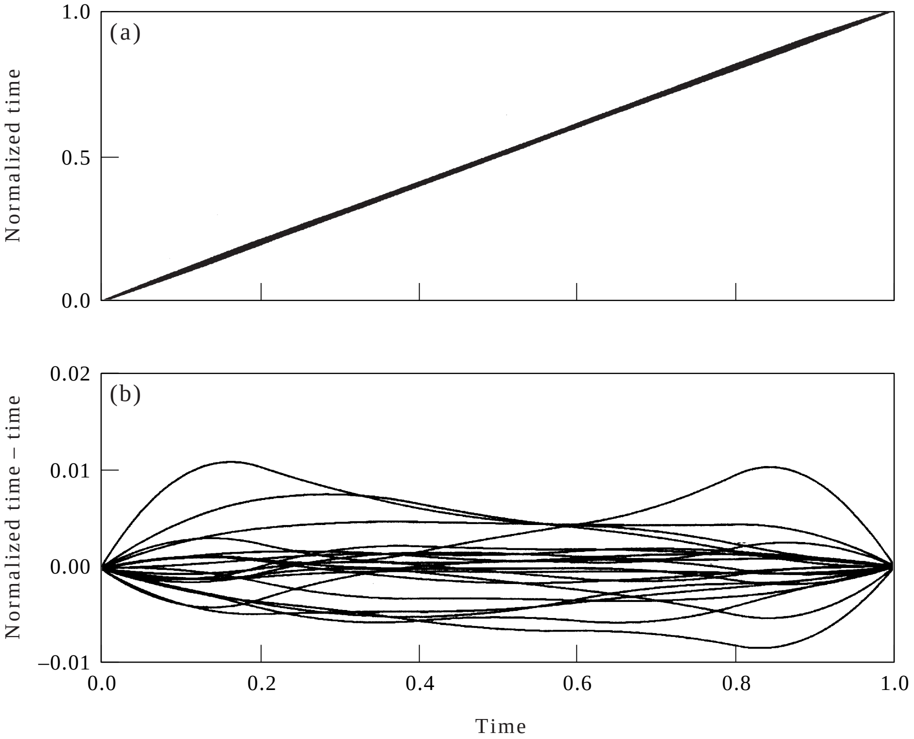

Figure 4. (a) Warping functions $h_i(t)$ and (b) difference $(h_i(t) - t)$.

The higher the value of W, the larger the phasing variability of the wavelets. In the present example, $W = 0{\cdot}0027$.

After the time normalization, the harmonic and noise components of the wavelets were extracted as the average, and the difference of each wavelet to the average, respectively. Figure 5 shows the harmonic component (top) and the noise components (middle). It also shows the standard deviation of the noise components (bottom). One can note that the noise is regularly distributed over most of the wavelet period, except for an increase at the final portion. The HNR was finally computed as defined by equation (8), with a result HNR = 325·33 (25·12 dB). For reference, the HNR computed by zero-padding the original wavelets is HNR = 52·03 (17·16 dB), and with linear interpolation to a common length is HNR = 87·44 (19·42 dB).

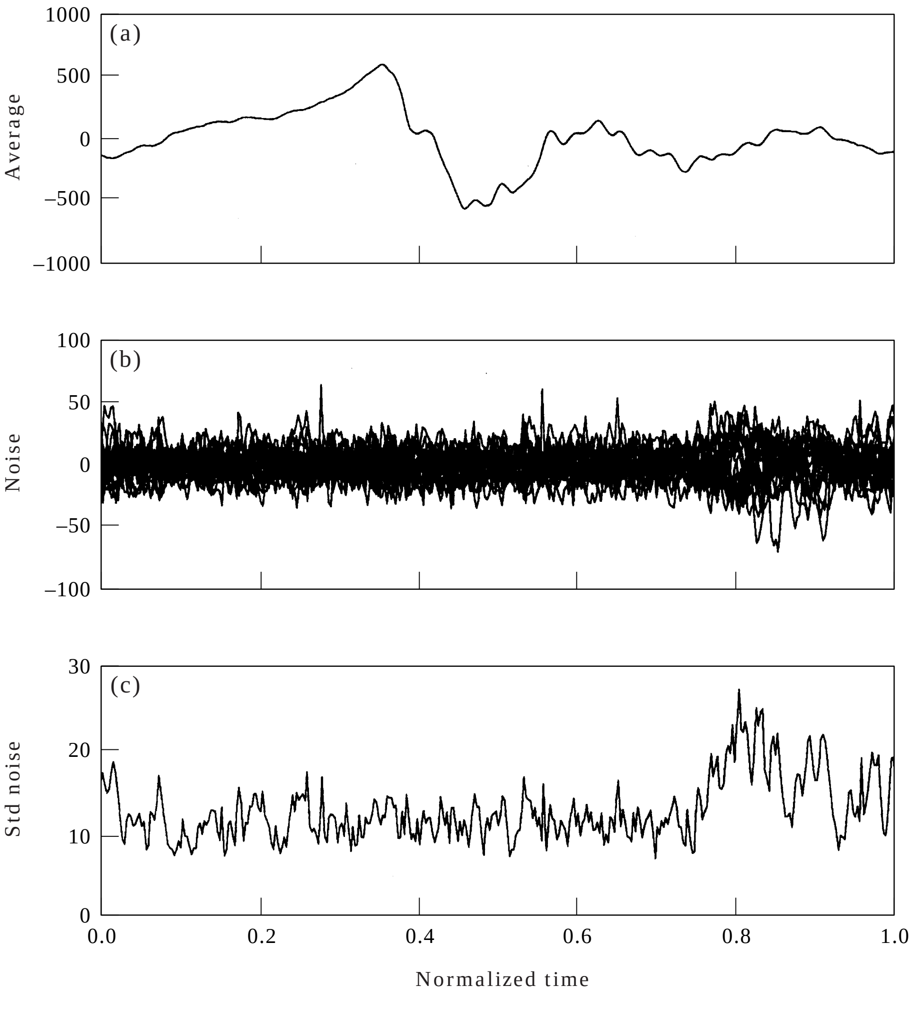

Figure 5. (a) Harmonic component, (b) noise components, and (c) standard deviation of noise components.

4. CONCLUSIONS

This letter has shown a FDA algorithm for the time normalization of voice signals, which might have more flexibility than previous dynamic programming approaches. The FDA algorithm permits the use of different optimization criteria according to the particular application. For example, one can use a weighted combination of derivatives of the wavelets, or time-dependent weight functions to emphasize portions of the time interval [7]. The resultant warping functions are smooth and differentiable, and might then be used for further analysis (as they contain information on the phasing variability of the wavelets). Further, there is no need to select a reference wavelet to serve as a template for the normalization.

The algorithm might have a wide application to investigate patterns and variability of sets of wavelets in various fields of knowledge. It may also be easily extended to vector-valued functions, for application to the simultaneous time normalization of multiple sets of wavelets.

ACKNOWLEDGEMENTS

I am grateful to Dr Janete M. Nicola for her interest and help to collect the voice signal. This work was supported by a Research Fellowship from the National Council for Scientific and Technological Development of Brazil (CNPq).

REFERENCES

1. E. Yumoto, W. J. Gould and T. Baer 1982 *Journal of the Acoustical Society of America* **71**, 1544–1550. Harmonics-to-noise ratio as an index of degree of hoarseness.
2. Y. Qi 1992 *Journal of the Acoustical Society of America* **92**, 2569–2576. Time normalization in voice analysis.
3. Y. Qi, B. Weinberg, N. Bi and W. J. Hess 1995 *Journal of the Acoustical Society of America* **97**, 2525–2532. Minimizing the effect of period determination on the computation of amplitude perturbation in voice.
4. Y. Qi and R. E. Hillman 1997 *Journal of the Acoustical Society of America* **102**, 537–543. Temporal and spectral estimations of harmonics-to-noise ratio in human voice signals.
5. J. C. Lucero, K. G. Munhall, V. L. Gracco and J. O. Ramsay 1997 *Journal of Speech, Language, and Hearing Research* **40**, 1111–1117. On the registration of time and the patterning of speech movements.
6. J. O. Ramsay and B. W. Silverman 1997 *Functional Data Analysis.* New York: Springer-Verlag.
7. J. O. Ramsay and X. Li 1998 *Journal of the Royal Statistical Society B* **60**, 351–363. Curve registration.
8. J. O. Ramsay 1998 *Journal of the Royal Statistical Society B* **60**, 365–375. Estimating smooth monotone functions.
9. K. Wang and T. Gasser 1997 *Annals of Statistics* **25**, 1251–1276. Alignment of curves by dynamic time warping.
10. I. R. Titze and H Liang 1993 *Journal of Speech and Hearing Research* **36**, 1120–1133. Comparison of F_0 extraction methods for high-precision voice perturbation measurements.

11. W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery 1992 *Numerical Recipes in C—The Art of Scientific Computing*. Cambridge: Cambridge University Press.