Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

Atmospheric Measurement Techniques Discussions

Computational Efficiency for the Surface Renewal Method

Jason Kelley and Chad Higgins

Dept. of Biological and Ecological Engineering, Oregon State University
116 Gilmore Hall, Corvallis, OR 97333 USA

Correspondence to: Jason Kelley (kelleyja@oregonstate.edu)

Abstract. Measuring surface fluxes using the surface renewal (SR) method requires a set of programmatic algorithms for tabulation, algebraic calculation, and quality control. Field experiments were conducted over a variety of surface conditions to determine the shortest valid time averaging period for the SR method. Because the SR method utilizes high frequency (20 Hz+) measurements, some steps in the flux calculation are computationally expensive, especially when automating SR to perform many iterations of these calculations. For a study on the minimum time to measure flux, several new algorithms were written to perform the required calculations more efficiently and rapidly. These algorithms demonstrate that simple modifications to SR methods can dramatically improve efficiency. Such programming techniques grant a degree of flexibility to the method, opening new avenues for SR to be used in research, for applied monitoring, and in novel field deployments.

1 Introduction

Eddy covariance (EC) has been established as a robust and accurate method to measure flux (Baldocchi, 2014), yet the surface renewal (SR) method offers several advantages over EC. While EC requires fast (10 Hz+) measurement of both the vertical wind speed and air temperature to measure the sensible heat flux, the SR method does not explicitly require vertical wind speed, determining flux from rapid temperature measurements alone. By requiring fewer sensors, cost is reduced, which expands direct flux measurement from research applications to more general applied monitoring (Paw U et al., 2005a; Spano et al., 2000). Another advantage of SR over methods such as EC is that SR can measure flux very near the surface or near the top of the plant canopy (Katul et al., 1996; Paw U et al., 1992). By taking measurements very close to the surface, the measurement fetch is reduced and the "flux footprint" is smaller (Castellví, 2012), yielding a more localized flux estimate.

The low cost of using SR to measure sensible heat flux is often cited as motivating factor is developing the method (Paw U et al., 2005a). By reducing sensor cost, multiple SR systems can be implemented to measure flux more extensively than a single EC sensor. Extensive, site specific SR estimates can augment the utility of sparsely located, permanent weather stations in mapping the heterogeneity of surface flux. Low cost sensors also facilitate measuring flux in situations where deploying more expensive equipment would be infeasible. Examples of situations which could benefit from low cost flux measurements include direct crop ET monitoring, experiments at remote field sites, and in developing regions. While many of these potential applications would be made possible by SR, the method still requires standardized calibration and quality control measures to establish that SR is robust, uniform, and accurate.

Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

Atmospheric Measurement Techniques Discussions

As described by (Van Atta, 1977), small scale turbulent transport primarily occurs during rapid events which manifest as ramp-like coherent structures. Because of the short duration of these events, SR potentially can estimate flux over shorter periods than the typical 15-30 minute averaging time used for EC. Rapid flux measurement will facilitate new applications, such as spatial mapping of flux using vehicle mounted, near surface sensors. Such a mobile SR implementation could provide new insights into the complexities of sub-basin scale hydrology, be used to validate downscaled models, and measure the heterogeneity of flux at sub-field scales. The implementation of SR on a moving vehicle to map flux will require finding the minimum time period in which a flux measurement can be obtained reliably. To find the minimum measurement period, field studies were conducted in 2014 and 2015 over various types of surface conditions. Fluxes were computed over a range of different time averaging periods with co-located EC and SR sensors (manuscript forthcoming). Initial attempts to calculate flux followed methods as described by Paw U et al. (2005a) and Snyder et al. (2008). However, implementing these methods literally as documented was hampered by slow computation time, which constrained the many iterations required to determine the minimum flux averaging period.

Open source software and online forums are abound with efficient methods that utilize advances in computing power, memory availability, and the accessibility of multithreaded processing. These methods reduce computational overhead, and can augment the SR technique as to allow implementation with low cost computers and data loggers, or where remote telemetry is required. In this manuscript, three example methods are shown which streamline specific operational steps in the SR method. The first is a rapid method to "despike" noisy data, a quality control technique commonly used in processing raw meteorological data. Second is a method to compute structure functions over multiple time lags rapidly using convolution. Third, an algebraic solution is used for array calculations to find roots of cubic polynomials, simplifying the determination of SR ramp amplitude. By using more efficient algorithms, many rapid iterative trials could be conducted to test hypotheses on the time averaging of flux calculations.

2 Methods

These algorithms adapt previously described methods, including despiking of time series data (Højstrup, 1993), calculation of structure functions (Antonia and Van Atta, 1978), and Fourier analysis of signals i.e. spectral analysis (Press, 2007; Stull, 1988). In each case, dramatically faster execution times were accomplished using simple programming improvements. Most efficiency gains were a result of code vectorization, which is the conversion of iterative looping algorithms into array calculations. All methods described here were implemented in the Matlab language (The Mathworks Inc., 2016), with the Statistics, Curve Fitting, and Signal Analysis Toolboxes. Matlab's Profiler (*profile.m*) was used to track the memory demand and speed of implementation. Trials were conducted on multiple desktop systems; analysis shown here only used test runs that were conducted on a Windows 10 operating system running on an Intel Core (TM) i7-3720QM processor operating at 2.60GHz with 16GB RAM. Processor clock speed was verified using Matlab's Profiler tool at run time, and processing times reported are described as either run time (actual observed execution time) or as Total Run Time, which is the sum of CPU time for all calculation threads. Example methods are indicated by *function name in italics*: abbreviated, commented code for

the example functions are provided in the supplementary material. Example data was collected during two field experiments in 2014 and 2015 using an integrated sonic anemometer and infrared gas analyzer (IRGASON) and fine wire thermocouples (FWTs), recorded at 10, 20, and 100 Hz using a CR1000 datalogger (Campbell Scientific). Complete description of the field experiments are presented in a forthcoming manuscript.

Vectorization of Matlab code entails removing loops (which are not pre-compiled) and taking advantage of implicit parallel methods in Matlab's pre-compiled functions (Altman, 2015). Other significant improvements were enabled through the Fast Fourier Transform by using convolution of number arrays, rather than iterative operations. In the case of determining ramp geometry in the SR method, Cardano's solution for depressed cubic polynomials (published in 1545) reduces a root finding algorithm from an iterative numerical approximation to an exact algebraic vector calculation. While some of the implementation of these methods are particular to the Matlab language, the general mathematical concepts are universal. Although these methods were prototyped in Matlab, the examples shown are generally useful as solutions to challenges commonly encountered in micrometeorology.

2.1 Despiking of noisy data using convolution

Despiking is the removal of erroneous or extreme data points from a time series of sampled values. It is a common procedure when measuring environmental parameters, especially in challenging conditions or complex environments (Göckede et al., 2004). The origin of spikes in a time series may be or electronic or physical (sensor malfunction or actual physical non-errors). Regardless of the origin, spikes can be recorded as abnormally large or small values, or may be marked by a firmware defined error flag. Spikes become problematic if they are not readily differentiated during automatic data imports (Rebmann et al., 2012). Spikes interfere with statistical calculations, and require some deliberate and objective method to identify, remove, and interpolate where they exist. For instance, a data logger program may record an error as "9999" or a character string, while Matlab denotes missing values in a numerical array as NaN ("not a number"). Because normally distributed data may contain noise in a wide range of values, robustness of the despiking algorithm is complicated by the requirement to differentiate between hard spikes characteristic of automatic flags (such as 9999) and soft spikes which are realistically valued but record erroneous measurement. An objective limit for soft spikes is usually defined as appropriate for the signal to noise ratio of any particular data, usually in terms of variance during some defined period. Clearly distinguishing errors can be achieved by a static objective criteria, by a dynamic statistic, or in a separate pre-processing operation. Previous authors have described a variety of methods including use of autocorrelation (Højstrup, 1993) and statistics within a moving window (Vickers and Mahrt, 1997).

Despiking is largely a problem of low pass filtering; consequently this procedure can be treated as a application of signal conditioning and therefore performed efficiently using convolution. Mathematically, convolution can be understood as a multiplicative function that combines a data signal with a filter signal. For a discrete signal, the filter is a weight array which is multiplied (in the Fourier domain) with data inside a window. In the time domain, the window can be visualized as moving along the data array as it is multiplied. As examples, a filter with a weight of 2 at the center of the window, and zeros elsewhere, would amplify the data signal by a factor of two; a filter 10 samples wide, each weighted at 0.1, generates a running average of the data. The computational efficiency of

Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

Atmospheric Measurement Techniques Discussions

convolution is a product of the Fast Fourier Transform (FFT), which allocates memory efficiently by a process known as bit switching. A thorough treatment of bit switching can be found in Chapters 12 and 13 of Press (2007). To demonstrate the increased efficiency of the FFT, two programs were used to despike 8.5 hours of 20 Hz sonic temperature data (609139 samples). A first attempt utilized a *for* loop, following the objective criteria described by Vickers and Mahrt (1997). A second program (*despike.m*) used convolution to determine a running mean and standard deviation used in the identification of spikes. After multiple runs with different input criteria, the first program average run time was 27 seconds. Using convolution, the second program average run time was 0.2 seconds, decreasing run time by approximately 99%. While this drastic improvement may overemphasize the slow compile times of loops in Matlab compared to other languages, it nonetheless demonstrates the value of the FFT in speeding signal conditioning, and facilitated the abandonment of time saving strategies sometimes used to speed despiking, such as skipping samples or limiting window size in calculating statistical moments.

To test computation time uniformly, identical 10 Hz data was sub-sampled to record lengths of 0.25 to 48 hours, and multiple runs were despiked with each sample set. Raw data was checked for hard error flags which required text to number conversion, but was not otherwise manipulated prior to despiking. Matlab Profiler was used to track the run time for all threads, using the undocumented flag "built-in" to track pre-compiled Matlab functions as well as user functions[1]. The total run time for all threads was tabulated and averaged across sets of each data length (Figure 1). By using convolution, despiking was two orders of magnitude faster for all lengths of data. To illustrate the effect of Matlab's built-in parallel processes, Figure 2 shows the ratio of actual run time to Total Run Time, indicating that the convolution method relies on computations conducted in parallel for processing increasingly longer data records. This benefit is direct accrued from the efficiency of the FFT.

2.2 Structure function calculation

Another computationally intensive process in SR is the determination of the 2nd, 3rd, and 5th order structure functions. Ramps are an identifiable feature in the measured temperature trace above any natural surface, yet determining the characteristic ramp geometry from high frequency data requires an efficient, robust, and preferably automated procedure. There are several methods to determine ramp geometry, including visual detection (Shaw and Gao, 1989), low pass filtering (Katul et al., 1996; Paw U et al., 1995), wavelet analysis (Gao and Li, 1993), and structure functions (Spano et al., 1997). Structure functions in particular provide both objective criteria to detect ramps and an efficient method to tabulate statistics of time series data. The general form for a structure function is:

$$S^n(r) = \frac{1}{N-r}\sum_{1}^{i=N-r}\left[T(i+r)-T(i)\right]^n \quad (1)$$

in which a vector of length N-1 is composed of differences between sequential samples of T(i), separated by lag r. The structure function $S^n(r)$ of order n for a given sample lag r is obtained by raising the difference vector to the n power, summing the vector and normalizing by N-1. The fluctuations of time series T(t) in a turbulent flow is a combination of random fluctuations and coherent structures (Van Atta and Park, 1972). The random (incoherent)

[1] http://undocumentedmatlab.com/blog/undocumented-profiler-options-part-4, Accessed January 2017

Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

Atmospheric Measurement Techniques Discussions

part of the signal is a product by isotropic turbulent processes, and over an adequately large sample this sample should have no particular sense (direction). The coherent structures generate anisotropic signatures, with periods of gradually changing temperature punctuated by sharp transitions during sweeps and ejections. Structure functions can be used to decompose the time series into isotropic and anisotropic components and identify the geometry of coherent structures (Van Atta, 1977).

Advances in sensor response time and processor speed have revealed an increasingly detailed picture of the coherent structures. In deriving a method to find ramp geometry, Van Atta (1977) calculated structure functions for eight different lags. Two decades later, increased processor power and memory size allowed Snyder et al., (1996) to calculate structure functions on 8 Hz data for lags from 0.25-1.0 seconds, but they were unable to resolve fluxes accurately at some measurement heights and surface roughness conditions. Later it was realized that determining the contributions from "imperfect ramp geometry" would require more thorough examination of ramp durations (Chen et al., 1997a; Paw U et al., 2005a). For this analysis of minimum SR averaging period, data was collected over periods of 8 hours to two months at sampling frequencies up to 100Hz. Initially, computation of structure functions for 3 minute periods with lags up to 10 seconds required 39 seconds on average with a series of nested loops. In contrast, using the convolution method presented below, this same calculation was accomplished in 7.6 seconds, an ~80% reduction in execution time. The function *strfnc.m* simultaneously time stamps the averaging period, finds the sign of $S^3(r)$ (used to find flux direction), and indexes the maximized value of $S^3(r)/r$, preparing the data for subsequent steps in determining flux. Using 100 Hz FWT data increased processing time using the convolution method to 38.4 seconds. The loop method would be unable to process 100 Hz data in real time applications, and would require long calculation time when using large numbers of records.

For a total of N sample lags, two dimensional convolution is performed using a filter matrix which is composed of N column vectors of length N+1: [1 -1 0 0...0], [1 0 -1 0 ...0],...[1 0 0 0.... -1]. Each column represents a single sample lag. When the filter matrix is convolved with time series data, the column vectors of the resulting matrix are difference vectors (*T(i+r) - T(i)*) as in Eq. 1, for each sample lag in the filter. Trials of 10Hz data using Matlab's Profiler showed that calculation efficiency is not accrued directly from convolution, but by changing the order of implementation. In the looping method, exponentiation (n = 2,3,5) is conducted on the difference vectors for each lag separately. The accelerated exponentiation in *strfnc.m* is possible by using matrix multiplication on the convolved matrix, and is faster due to compact memory allocation of the FFT. The resulting efficiency is not dependant on total data size, but is strongly dependent on the length of the averaging period used to calculate flux. This is a critical improvement for iterative SR calculations used to minimize the SR measurement period (Figure 3).

Because the SR calculation must identify the lag which maximizes the ratio $S^3(r)/r$, structure functions must be continuously calculated for a range of lagged differences up to the expected time scale of coherent structures. A short maximum lag (3-5 seconds) may be adequate under some atmospheric stability conditions in which buoyancy drives temperature flux via rapid exchange rates. Under stable conditions, though, longer lags may be important to detect the true maximum of the ratio $S^3(r)/r$, indicating the longest time scale contributing to flux. To evaluate sensitivity in response to the number of calculated lags, the structure functions were calculated iteratively, varying the averaging period and maximum tested lag time (Figure 4). For all maximum lags, the convolution method was

Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

Atmospheric Measurement Techniques Discussions

between 4 and 14 times faster than the loop method, with short averaging periods again the largest factor in the difference between the two methods.

2.3 Cardano's method for depressed cubic polynomials

For the SR method, the structure functions are retained from the lag which maximizes the ratio $S^3(r)/r$. The resulting values are used as coefficients in a cubic polynomial, the root of which is the ramp amplitude (A) used to calculate flux:

$$A^3 + \left(10S^2(r) - \frac{S^5(r)}{S^3(r)}\right)A + 10S^3(r) = 0 \quad (2)$$

The magnitude of the real root (of 3 possible roots) is regarded as the corresponding solution for ramp amplitude (Spano et al., 1997). The Matlab root finding algorithm computes eigenvalues of a companion matrix to approximate the solution to a n^{th} order polynomial, regarding the input function as a vector with n+1 elements (*roots.m* documentation[2]). Consequently, this function cannot be executed directly on an array. The algebraic solution method proposed by Gerolamo Cardano in the 1545 *Ars Magna* for "depressed" cubics, on the other hand, can be applied to vectors. Cardano's solution is found by substituting A with $(m^{1/3} + n^{1/3})$ into the abbreviated equation $A^3+pA+q=0$. Expanding terms and using the quadratic equation yields an exact solution:

$$A = \left(-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}\right)^{\frac{1}{3}} + \left(-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}\right)^{\frac{1}{3}} \quad (3)$$

where p and q are the coefficients of the depressed cubic and derived from the structure functions (Edwards and Beaver, 2015). The function *cardanos.m* was adapted from a function by Bruno Luong[3], in a reduced form for the real valued cases used to implement the SR method. The function output was verified against the Matlab function *roots.m* for polynomials with both positive and negative real valued inputs (imaginary inputs are irrelevant for SR flux measurements). Solution for the real roots in this manner expedites determination of flux magnitude and direction. The algebraic root finding method simplifies and speeds iterative application of the SR method by operating directly on arrays.

Solving for the roots of this function yields a single, predominant ramp amplitude from a given temperature trace (with units of °C or K). In addition to ramp amplitude, the time scale or ramp duration must also be determined. (Van Atta, 1977) suggested that ramp time τ should be related linearly to amplitude A, and proposed:

$$\tau = \frac{-A^3 r}{S^3(r)} \quad (4)$$

[2] http://www.mathworks.com/help/matlab/ref/roots.html, Accessed 9AUG2016

[3] https://www.mathworks.com/matlabcentral/newsreader/view_thread/165013, Accessed 10MAY2016

In practice, determination of ramp time τ from A using this equation requires an empirical calibration; this calibration has been shown to be related to surface conditions and instrumentation (Chen et al., 1997b). Ongoing work using replicate measurements at multiple heights (Castellvi, 2004) and frequency response calibration (Shapland et al., 2014) have begun to resolve the causes of variability in this parameter. In this study, it was found that the ratio in equation 4 remains essentially constant for a given surface roughness condition, allowing determination of τ algebraically. Automated computation of equation 2 using the exact solution facilitates rapid evaluation of the ramp geometry, and determining flux magnitude from ramp geometry is a relatively simple matter of linear scaling when calibrating to a control measure such as eddy covariance.

3 Conclusions

As with other methods for measuring flux from the surface, analytic solutions do not always translate easily into straightforward numerical computation, especially when working with long term and high frequency records. In applied research, custom algorithms are often developed by individual researchers, requiring special training in programming, significant time investment, and the motivation to use sophisticated techniques that fully utilize available memory and processing power. Efforts to standardize the eddy covariance method (Aubinet et al., 2012; Baldocchi, 2014) and data quality control (Allen et al., 2011; Foken et al., 2012) have not yet been similarly applied to the SR method, although substantial work has been made to validate calibration and field methods (Castellví, 2012; Paw U et al., 2005b; Shapland et al., 2014). By appropriating methods common in signal processing and computer science, and by sharing open source tools using online forums such as stackexchange.com, more sophisticated approaches can be implemented. One goal of these open source efforts should be to reduce the computational overhead of calculating flux, so as to facilitate the broadest implementation of the SR method in applied contexts. Reducing the cost and power requirement of the required data loggers, computers, and telemetry will facilitate the extensive deployment of SR sensors to aid in describing the heterogeneity of flux across the landscape.

4 Data Availability

All data used in this analysis and scripts implementing the algorithms described above are available online at http://hdl.handle.net/1957/60599.

Abbreviated scripts for the three example methods may be found in the supplemental materials.

References

Allen, R., Pereira, L. S., Howell, T. A. and Jensen, M. E.: Evapotranspiration information reporting: I. Factors governing measurement accuracy, Agricultural Water Management, 98(6), 899–920, 2011.

Altman, Y.: Accelerating Matlab Performance, CRC Press., 2015.

Antonia, R. A. and Van Atta, C. W.: Structure functions of temperature fluctuations in turbulent shear flows, Journal of Fluid Mechanics, 84(03), 561–580, 1978.

Aubinet, M., Vesala, T. and Papale, D.: Eddy covariance: a practical guide to measurement and data analysis, Springer Science & Business Media., 2012.

Baldocchi, D.: Measuring fluxes of trace gases and energy between ecosystems and the atmosphere–the state and future of the eddy covariance method, Global change biology, 20(12), 3600–3609, 2014.

Castellvi, F.: Combining surface renewal analysis and similarity theory: a new approach for estimating sensible heat flux, Water resources research, 40(5), 2004.

Castellví, F.: Fetch requirements using surface renewal analysis for estimating scalar surface fluxes from measurements in the inertial sublayer, Agricultural and Forest Meteorology, 152, 233–239, doi:10.1016/j.agrformet.2011.10.004, 2012.

Chen, W., Novak, M., Black, T. A. and Lee, X.: Coherent eddies and temperature structure functions for three contrasting surfaces. Part I: Ramp model with finite microfront time, Boundary-Layer Meteorology, 84(1), 99–124, doi:10.1023/A:1000338817250, 1997a.

Chen, W., Novak, M., Black, T. A. and Lee, X.: Coherent eddies and temperature structure functions for three contrasting surfaces. Part II: Renewal model for sensible heat flux, Boundary-Layer Meteorology, 84(1), 125–147, doi:10.1023/A:1000342918158, 1997b.

Edwards, A. C. and Beaver, J. M.: Investigating Cardano's Irreducible Case, 2015 NCUR, 2015.

Foken, T., Leuning, R., Oncley, S. R., Mauder, M. and Aubinet, M.: Corrections and data quality control, in Eddy Covariance, pp. 85–131, Springer., 2012.

Gao, W. and Li, B. L.: Wavelet analysis of coherent structures at the atmosphere-forest interface, Journal of Applied Meteorology, 32(11), 1717–1725, 1993.

Göckede, M., Rebmann, C. and Foken, T.: A combination of quality assessment tools for eddy covariance measurements with footprint modelling for the characterisation of complex sites, Agricultural and Forest Meteorology, 127(3), 175–188, 2004.

Højstrup, J.: A statistical data screening procedure, Measurement Science and Technology, 4(2), 153, 1993.

Katul, G., Hsieh, C.-I., Oren, R., Ellsworth, D. and Phillips, N.: Latent and sensible heat flux predictions from a uniform pine forest using surface renewal and flux variance methods, Boundary-Layer Meteorology, 80(3), 249–282, 1996.

Paw U, K. T., Brunet, Y., Collineau, S., Shaw, R. H., Maitani, T., Qiu, J. and Hipps, L.: On coherent structures in turbulence above and within agricultural plant canopies, Agricultural and Forest Meteorology, 61(1–2), 55–68, doi:10.1016/0168-1923(92)90025-Y, 1992.

Paw U, K. T., Qiu, J., Su, H.-B., Watanabe, T. and Brunet, Y.: Surface renewal analysis: a new method to obtain scalar fluxes, Agricultural and Forest Meteorology, 74(1–2), 119–137, doi:10.1016/0168-1923(94)02182-J, 1995.

Paw U, K. T., Snyder, R. L., Spano, D. and Su, H.-B.: Surface Renewal Estimates of Scalar Exchange, in Micrometeorology in Agricultural Systems, pp. 455–483, American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI. [online] Available from: http://dx.doi.org/10.2134/agronmonogr47.c20, 2005a.

Paw U, K. T., Snyder, R. L., Spano, D. and Su, H.-B.: Surface Renewal Estimates of Scalar Exchange, in Micrometeorology in Agricultural Systems, pp. 455–483, American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI. [online] Available from: http://dx.doi.org/10.2134/agronmonogr47.c20, 2005b.

Press, W. H., Ed.: Numerical recipes: the art of scientific computing, 3rd ed., Cambridge University Press, Cambridge, UK ; New York., 2007.

Rebmann, C., Kolle, O., Heinesch, B., Queck, R., Ibrom, A. and Aubinet, M.: Data acquisition and flux calculations, in Eddy Covariance, pp. 59–83, Springer., 2012.

275 Shapland, T. M., Snyder, R. L., Paw U, K. T. and McElrone, A. J.: Thermocouple frequency response compensation leads to convergence of the surface renewal alpha calibration, Agricultural and Forest Meteorology, 189–190, 36–47, doi:10.1016/j.agrformet.2014.01.008, 2014.

Shaw, R. H. and Gao, W.: Detection of temperature ramps and flow structures at a deciduous forest site, Agricultural and forest meteorology, 47(2), 123–138, 1989.

280 Snyder, R. L., Spano, D. and Pawu, K. T.: Surface renewal analysis for sensible and latent heat flux density, Boundary-Layer Meteorology, 77(3–4), 249–266, 1996.

Snyder, R. L., Spano, D., Duce, P., Paw U, K. T. and Rivera, M.: Surface renewal estimation of pasture evapotranspiration, Journal of Irrigation and Drainage Engineering, 134(6), 716–721, 2008.

Spano, D., Snyder, R. L., Duce, P. and Paw U, K. T.: Surface renewal analysis for sensible heat flux density using
285 structure functions, Agricultural and Forest Meteorology, 86(3), 259–271, 1997.

Spano, D., Snyder, R. L. and Duce, P.: Estimating sensible and latent heat flux densities from grapevine canopies using surface renewal, Agricultural and Forest Meteorology, 104(3), 171–183, 2000.

Stull, R. B.: An introduction to boundary layer meteorology, Springer., 1988.

The Mathworks Inc.: Matlab R2016b, The MathWorks Inc., Natick, MA., 2016.

290 Van Atta, C. W.: Effect of coherent structures on structure functions of temperature in the atmospheric boundary layer, Archiwum Mechaniki Stosowanej, 29(1), 161–171, 1977.

Van Atta, C. W. and Park, J.: Statistical self-similarity and inertial subrange turbulence, in Statistical Models and Turbulence, pp. 402–426, Springer., 1972.

Vickers, D. and Mahrt, L.: Quality control and flux sampling problems for tower and aircraft data, Journal of
295 Atmospheric and Oceanic Technology, 14(3), 512–526, 1997.

Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

Atmospheric Measurement Techniques Discussions

Figures

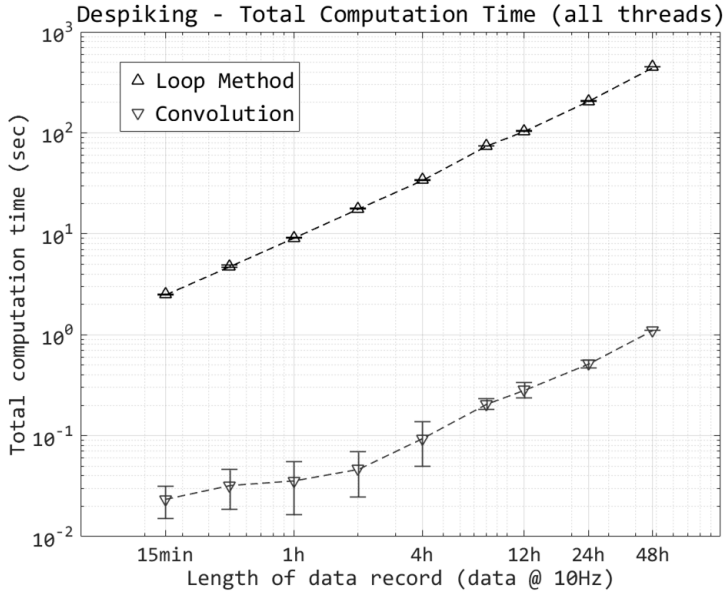

Figure 1: The total computation time is the sum of CPU time spent on all calculation threads. Marker is the mean run time for multiple runs, which varied from 30 runs (15min - 4h data) to 10 runs (8, 12, 24 h). 48h calculation is represented by one run only. Error bars represent one standard deviation of all runs.

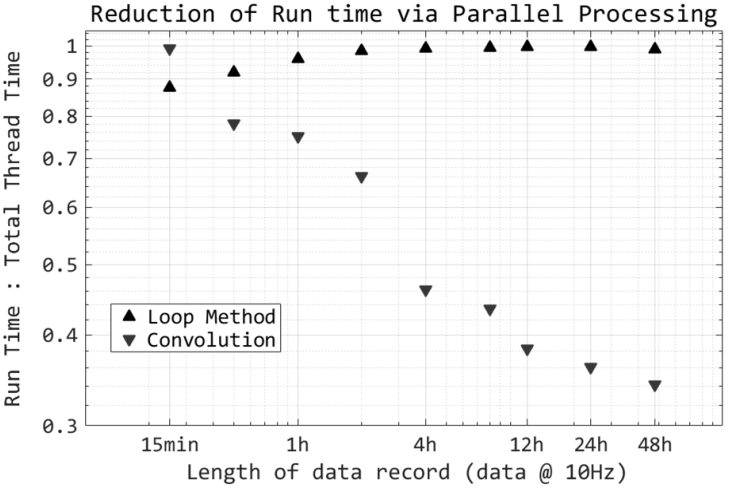

Figure 2: Fast calculation of larger data sets is due to implicit parallel processing via the FFT, which is readily performed by multiple simultaneous threads. The efficiency of parallel processing is shown by a lower ratio of Run Time to Total Thread Time.

Atmos. Meas. Tech. Discuss., doi:10.5194/amt-2017-123, 2017
Manuscript under review for journal Atmos. Meas. Tech.
Discussion started: 1 June 2017

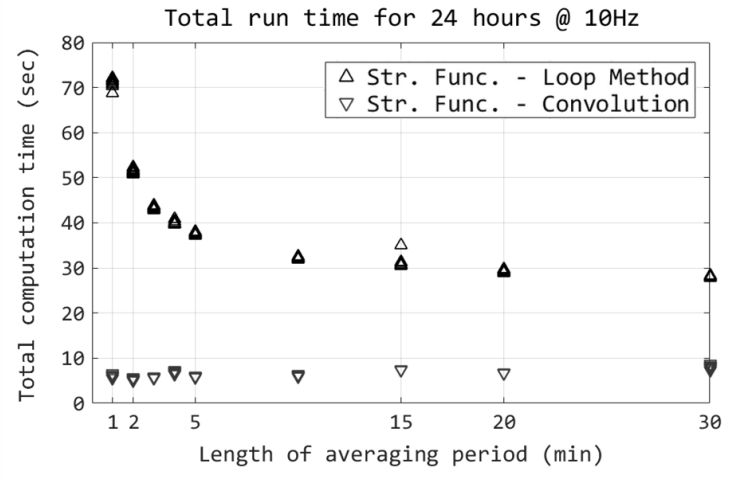

Figure 3: Ten iterations of the structure function calculations using a range of averaging periods.

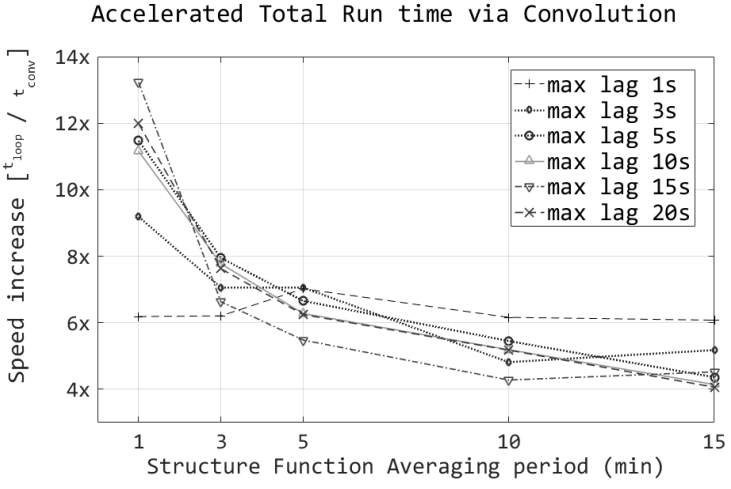

Figure 4: The performance gains using convolution are more significant for short averaging periods, regardless of maximum lag used in calculating structure functions.