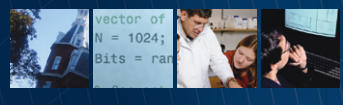

Using Information-Theoretic Statistics in MATLAB to Understand How Ecosystems Affect Regional Climates

By Benjamin L. Ruddell, Arizona State University, Tempe, AZ; Nils Oberg, Marcelo Garcia, and Praveen Kumar, University of Illinois at Urbana-Champaign, Urbana, IL

Products Used

- MATLAB®
- MATLAB Distributed Computing Server™
- Parallel Computing Toolbox™
- Statistics Toolbox™

IT IS COMMON KNOWLEDGE that weather and climate influence the plants, animals, and microorganisms that live upon the landscape. New research is investigating the possibility that the opposite is also true: that due to feedback between plants and the atmosphere, vegetation and the landscape influence regional climate.

University of Illinois at Urbana-Champaign researchers have developed statistical methods to detect connections between environmental variables, such as evaporation from plant leaves; isolate variables that drive changes in other variables; and identify feedback loops. The project is funded by NASA and the Metropolitan Water Reclamation District of Greater Chicago.

To understand the relationship between key variables in a self-organizing system, such as the Earth's land-surface ecosystem and climate, we must look beyond traditional linear methods of analysis. In a linear system, changes in subsystem "X" cause proportional changes in subsystem "Y". In a nonlinear or self-organizing system featuring circular feedback, this notion of causality breaks down, as components "X" and "Y" become self-causing. Furthermore, self-organizing feedback loops may be nested inside each other so that the system's dynamics behave like a Russian Doll, where each physical process is a small part of a larger feedback loop. This type of self-organizing system is best understood as a *process network*[1]. Process networks describe complex systems as networks of nested feedback loops and their associated timescales. Using a new class of advanced statistics based on the Theory of Statistical Information, process networks can be derived for any system that can be observed and measured.

Using MATLAB® and Parallel Computing Toolbox™, we apply these computationally intensive statistical methods to time series data, including observed meteorological, hydrological, and environmental variables. The results are helping to explain not only how changes in climate, including drought, affect the ecosystem, but also how human changes to landscape and vegetation affect the regional climate.

About the Authors

Benjamin Ruddell is an Assistant Professor in the Department of Engineering at Arizona State University. **Nils Oberg** is a Research Programmer, and **Marcelo Garcia** and **Praveen Kumar** are Professors in the Department of Civil Engineering at the University of Illinois at Urbana-Champaign. The work was performed at the Ven Te Chow Hydrosystems Lab at the University of Illinois.

[1] Ruddell and Kumar, 2009a

Tackling a Computationally Intensive Problem

The observed data is derived from FLUXNET, a global network of more than 400 towers, each equipped with a suite of sensors (Figure 1). These sensorsrecord air temperature (Θ_a), soil temperature (Θ_s), soil water content (θ), radiation from the sun (R_g), vapor pressure density (*VPD*, a measure of humidity), precipitation (P), cloud cover (C_F), the net flow of carbon dioxide in or out of the ecosystem (*NEE*), and the amount of heat radiated from the ground as sensible heat flux (γ_H) and as latent heat flux (γ_{LE}, evaporated water) (Figure 2). The measurements are averaged to a time resolution of 30-minute intervals. The Bondville tower used to study the structure of drought is located near Champaign, Illinois. This tower has been measuring the climate since 1996.

For each combination of two variables measured by the fluxnet tower, a joint probability distribution is estimated from the time series data. The information-theoretic statistic *transfer* entropy, which establishes statistically causal links between variables, requires estimation of a 3D joint probability density function. This computation must be repeated for all possible combinations of variables and time lags for each month of data studied. We can then examine how the process network of connections between variables changes with the seasons and understand the effects of drought on the structure of the system.

The computationally intensive nature of this approach was one of the main reasons we chose MATLAB. MATLAB is well-suited to the matrix manipulation required for the analysis, and Parallel Computing Toolbox enabled us to accelerate the computations by running them on

Figure 1. An eddy-covariance flux tower in Champaign, Illinois, near the Bondville site.

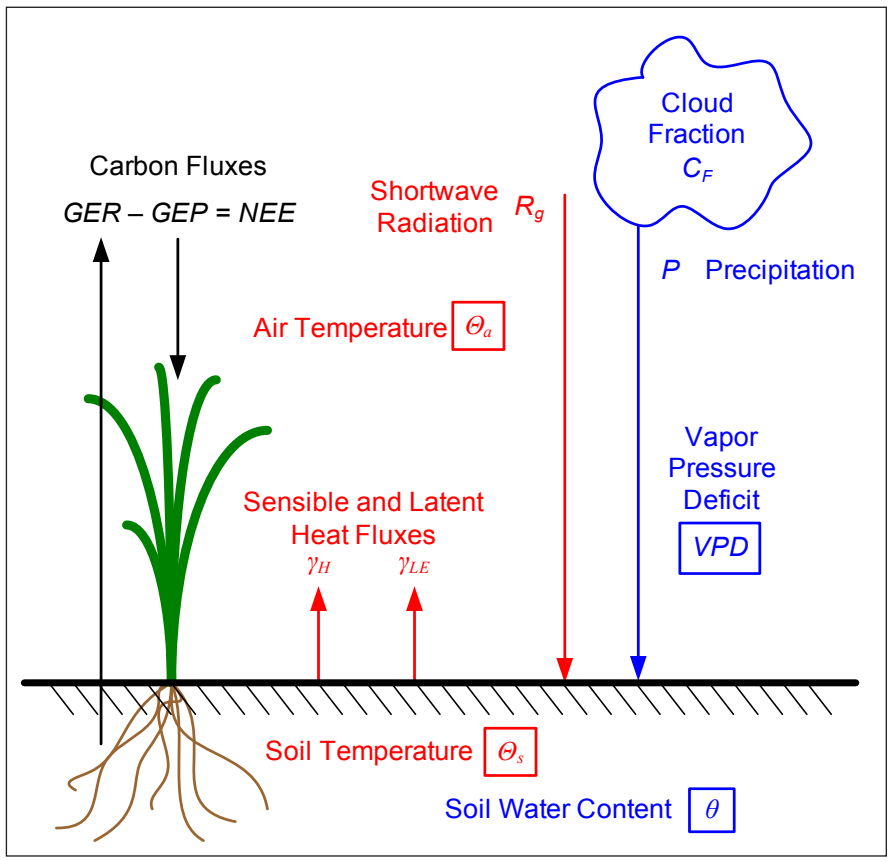

Figure 2. Variables used in the analysis.

a computing cluster. In addition, MATLAB visualization capabilities allowed us to rapidly analyze a large volume of statistical results.

The first step was to make sure that the data we received from the FLUXNET towers was complete and correctly formatted. Using MATLAB and Statistics Toolbox™, we wrote scripts to extract the subset of data that we needed, scan it for errors and omissions, fill in missing data when possible, and format the data for use in the statistical algorithms. Statistics Toolbox was used to summarize the input data set by month, season, and year to allow the plotting of results.

Estimating the transfer entropy statistic depends on the accurate estimation of probability densities from data. To calculate densities, we developed MATLAB algorithms for a fixed-interval partition (or bin-counting) classification scheme to estimate joint probabilities.

We obtained several interesting statistical results by applying transfer entropy to study the system's process network, including the *monthly mean net information production* of each variable. Information production measures the predictive value of each variable on the process network; a variable with a sufficiently large positive net information production causally drives other variables on the network more than it is driven by those variables (Figure 3). Due to feedback on the process network, all variables control the behavior of the network as a whole, but the variables marked red in Figure 3 have the greatest controlling influence.

Parallelizing the Application

During initial prototyping, the analysis focused on data from just two months at a single site. MATLAB algorithms were run on a dedicated workstation overnight because they took several hours to complete. When we began analyzing ten years' worth of data across multiple sites, we realized that the complete calculations would take about a month. This is too long to wait for results, especially when debugging and code alterations will necessitate multiple calculations.

Clearly, we would need to accelerate the analysis by parallelizing the algorithms and running them on a computer cluster. Fortunately, we can analyze the data set of each month and each tower site separately, making data analysis relatively easy to parallelize. However, there are always challenges to working in a cluster environment. When parallelizing a Fortran application, for example, developers may need to tailor it to account for cache and memory limitations, write initialization and staging scripts, and adapt the code to handle the unique properties of the cluster machines.

With MATLAB and Parallel Computing Toolbox, we parallelized our algorithm by changing a single line of code. In fact, the most difficult part of the

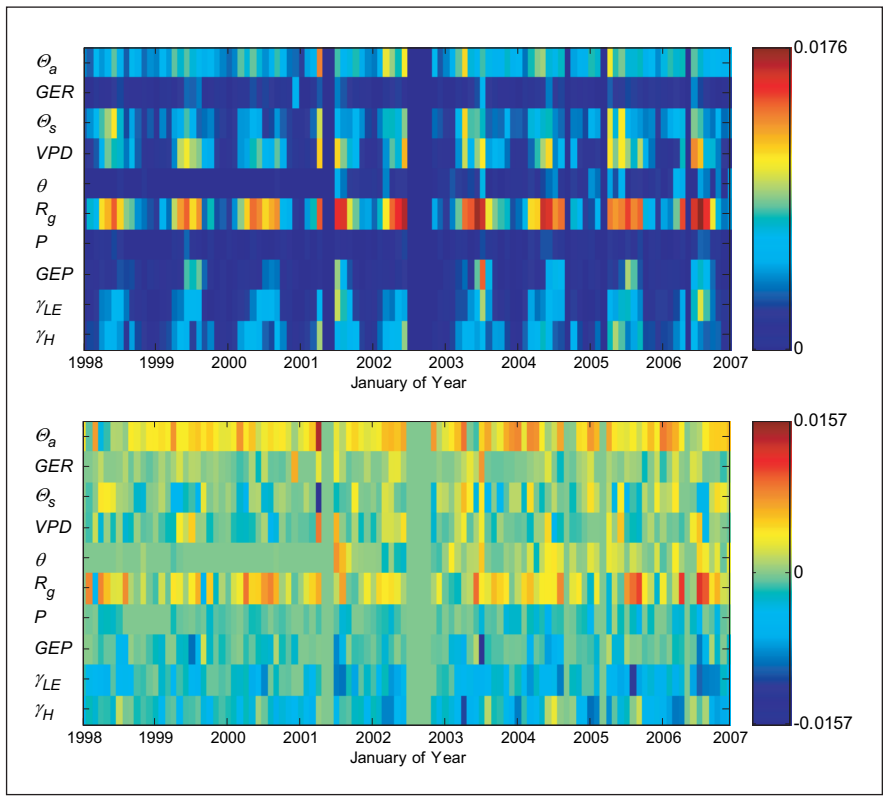

Figure 3. Mean gross (top) and net (bottom) information production for variables tracked at the Bondville, IL, site every month from 1998 to 2006. The visualization shows that the air temperature (Θ_a) and solar radiation (R_g) are strong causal drivers of the process network, especially during the summer (red indicates strong information production and therefore strong causal control over the system). Used with permission from Ruddell and Kumar (2009b).

parallelization procedure was convincing ourselves that one code modification—changing a `for` loop to a `parfor` (parallel `for`) loop—was all that was needed. The original code was not explicitly designed for parallelization, yet it took us less than an hour to convert the code to run in parallel on a computing cluster. Results computed by each "worker" were collected in a single six-dimensional array, which was then diced and visualized to display results.

The analysis was run on a 32-core cluster comprising four dual-CPU, quad-core systems. We saw a linear improvement in computation speed, completing in one day what would have required a month on a single workstation. A calculation with millions of iterations that took 176 hours on one core required just 5.46 hours using 32 cores.

Applying the Methods to Other Disciplines

Our research confirms that changes in the landscape and the ecosystem can affect regional climate via regional feedback loops in the process network. The implication of this finding is that, for example, land-use decisions can influence the severity and duration of droughts in the Midwestern U.S. Using this information it may be possible to design land-use policies for agriculture, forestry, and urban development that minimize adverse effects on regional climate.

We are collaborating with researchers who will apply these statistical methods to other time-varying complex systems in which feedback between component parts leads to self-organization. In one study, scientists are analyzing time series chemical concentrations in a closed system of microorganisms and nutrients to better understand the biological cycles involved. In another, researchers are analyzing satellite data to investigate the interaction of different parts of the landscape. Financial market analysis is another ideal application of this statistical approach.

Whatever the discipline, the algorithms that we use employ cutting-edge statistical methods and are exceptionally intensive computationally. MATLAB, Statistics Toolbox, and Parallel Computing Toolbox provide an advantage, both in the development of the algorithms and in the ability to use parallel computing to obtain and visualize results rapidly. ■

References

- Ruddell, B.L. and P. Kumar (2009a). "Ecohydrologic Process Networks: 1. Identification." *Water Resources Research* 45, W03419, doi:10.1029/2008WR007279.
- Ruddell, B.L. and P. Kumar (2009b). "Ecohydrologic Process Networks: 2. Analysis and Characterization." *Water Resources Research* 35, W03420, doi:10.1029/2008WR007280.

For More Information

- Demo: Benchmarking Distributed Jobs on the Cluster
  www.mathworks.com/benchmarking-parallel
- Demo: Simple Benchmarking of Parfor Performance Using Blackjack
  www.mathworks.com/benchmarking-parfor

Resources

VISIT
www.mathworks.com/academia

TECHNICAL SUPPORT
www.mathworks.com/support

ONLINE USER COMMUNITY
www.mathworks.com/matlabcentral

DEMOS
www.mathworks.com/demos

TRAINING SERVICES
www.mathworks.com/training

THIRD-PARTY PRODUCTS AND SERVICES
www.mathworks.com/connections

Worldwide CONTACTS
www.mathworks.com/contact

E-MAIL
info@mathworks.com

91803v00 02/10