



Jul 1st, 12:00 AM

Semantic Array Programming for Environmental Modelling: Application of the Mastrave Library

Daniele De Rigo

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

De Rigo, Daniele, "Semantic Array Programming for Environmental Modelling: Application of the Mastrave Library" (2012).
International Congress on Environmental Modelling and Software. 69.
<https://scholarsarchive.byu.edu/iemssconference/2012/Stream-B/69>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Semantic Array Programming for Environmental Modelling: Application of the Mastrave Library

Daniele de Rigo ^{a,b}

(a) Politecnico di Milano, Dipartimento di Elettronica e Informazione,
Via Ponzio 34/5, I-20133 Milano, Italy. derigo@elet.polimi.it

(b) Joint Research Centre of the European Commission,
Institute for Environment and Sustainability, Via Fermi, 2749, I-21027 Ispra, Italy.
daniele.de-rigo@jrc.ec.europa.eu

Abstract: Environmental datasets grow in size and specialization while models designed for local scale are often unsuitable at regional/continental scale. At regional scale, data are usually available as georeferenced collections of spatially distributed despite semantically atomic information. Complex data intrinsically impose modellers to manipulate nontrivial information structures. For example, multi-dimensional arrays of time series may be composed by slices of raster spatial matrices for each time step, whilst heterogeneous collections of uneven arrays are common when dealing with data analogous to precipitation events, and these structures may ask for integration at several spatial scales, projections and temporal extents. Interestingly, it might be far more difficult to practically implement such a complexity rather than conceptually describe it: a subset of modelling generalizations may deal more with abstraction rather than with the explosion of lines of code. Many environmental modelling algorithms are composed by chains of data-transformations or trees of domain specific sub-algorithms. Concisely expressing them without the need for paying attention on the enormous set of spatio-temporal details, is a highly recommendable practice in both mathematical formulation and implementation.

The use of *semantic array programming* paradigm is here exemplified as a powerful conceptual and practical (with the free software library Mastrave) tool for easing scalability and semantic integration in environmental modelling. Array programming, AP, is widely used for its computational effectiveness but often underexploited in reducing the gap between mathematical notation and algorithm implementations, i.e. by promoting arrays (vectors, matrices, tensors) as atomic quantities with extremely compact manipulating operators. Coherent array-based mathematical description of models can simplify complex algorithm prototyping while moving mathematical reasoning directly into the source code – because of its substantial size reduction – where the mathematical description is actually expressed in a completely formalized and reproducible way. The proposed paradigm suggests to complement the characteristic AP weak typing with semantics, both by composing generalized modular sub-models and via array oriented – thus concise – constraints. The Mastrave library use is exemplified with a regional scale benchmark application to local-average invariant (LAI) downscaling of climate raster data. Unnecessary errors frequently introduced by non-LAI up-sampling are shown to be easily detected and removed when the scientific modelling practice is terse enough to let mathematical reasoning and model coding merge together.

Keywords: environmental datasets; regional scale; data-transformation modelling; semantic array programming, local-average invariant downscaling.

1 INTRODUCTION

Environmental modelling is inherently a multidisciplinary scientific activity which needs the coordination of a wide range of competences. Data modelling and data integration in environmental sciences are actively and increasingly investigated within information and communications technology [Casagrandi and Guariso 2009]. Environmental datasets are growing in size and specialization [Edwards 2004; Hijmans *et al.* 2005; Yesson *et al.* 2007; Scholes *et al.* 2008] while models designed for local scales are often unable to

work at regional/continental scale¹ especially when local-scale data requirements exceed or are unsuitable to exploit data availability at larger scale [Merritt *et al.* 2003] – despite large-scale analysis may be essential for assessing, planning and managing natural resources [Loarie *et al.* 2009; Angelis-Dimakis *et al.* 2011]. At regional scale, data are usually available as geographically referred collections of information distributed in space and time, such as raster layers of matrices each element of them refers to a given time, spatial location and resolution, possibly along with corresponding spatially distributed reliability information. Complex data intrinsically impose modellers to manipulate non-trivial information structures. For example, multi-dimensional arrays of time series of measures, geographically referred, may be composed by slices of raster matrices for each time step [Haylock *et al.* 2008; van den Besselaar *et al.* 2011], whilst heterogeneous collections of uneven arrays are common when dealing with data analogous to precipitation events (e.g. derived from detailed time series of measured precipitation so to remove low- or zero-intensity periods [de Rigo and Bosco 2011]).

Many environmental modelling algorithms are composed by chains of data-transformation trees of often loosely coupled or orthogonal sub-algorithms, focused on domain-specific aspects whose interaction contributes to better constrain and characterize the modelled quantities [Crowley and Hyde 2008; Lovett *et al.* 2009; Haughton *et al.* 2009; Hashimoto *et al.* 2011]. Concisely expressing them without the need for paying attention on the enormous set of spatial details, is a highly recommendable practice in both mathematical formulation and implementation.

This paper exemplifies the use of the emerging paradigm of *semantic array programming* (SAP) [de Rigo 2012] as a powerful conceptual and practical tool (with the free software library Mastrave) for easing scalability and semantic integration within environmental modelling. The focus is toward facilitating the implementation and evolution of the mathematics underlying complex models by means of concise, abstract and semantically enhanced notations and tools. The Mastrave modelling library implements the SAP paradigm allowing derived domain-specific models to share common semantic notation and existing non-SAP models to be integrated through lightweight semantic enhancement so to strengthen their robustness in an unobtrusive way. An example of the mutual reinforcement between concise array-based notation and semantic constraints will be described, and a regional (continental) scale application to local-average invariant (LAI) downscaling of benchmark climate raster data will serve to further illustrate the approach.

1.1 The Mastrave Modelling Library

Mastrave² is a free software [Stallman 2009] library written to perform SAP and to be as compatible as possible with both GNU Octave [Eaton *et al.* 2008] and MATLAB³ computing frameworks. The GNU Bash shell [Ramey and Fox 2006] is also transparently integrated⁴, to take advantage of some of its relevant stable and almost universally portable features – which have been headed toward the AP paradigm.

Mastrave is mostly oriented to ease complex modelling tasks such as those typically needed within environmental models, even when involving irregular and heterogeneous data series. Since 2005, the Mastrave library supports designing and implementing environmental modelling applications. Examples of applications range from evolutionary techniques for non-trivial parameter training (the SIEVE architecture – Selective Improvement by Evolutionary Variance Extinction, applied to improve [de Rigo *et al.* 2005] approximate dynamic programming in water resources [de Rigo *et al.* 2001]), up to on-line policy design for water reservoir networks [Castelletti *et al.* 2008]; from a modelling architecture for evaluating at continental scale potential and actual soil water erosion [de Rigo and Bosco 2011; de Rigo and Bosco (in prep.); Bosco *et al.* (in prep.)], up to several forest resource applications, such as detailed European forest tree species distribution modelling [de Rigo *et al.* (in prep.)], concise graph-based formulation [Estreguil *et al.* (submitted)] and nonlinear statistical analysis of heterogeneous spatial pattern indices for characterizing forest habitats [Estreguil *et al.* (in prep.)].

1 An opposite phenomenon also affects modelling integration and robustness [Trivedi *et al.* 2008].

2 The Mastrave Project, <http://mastrave.org/>

3 The MathWorks: MATLAB, <http://www.mathworks.it/help/techdoc/>

4 Since other languages – not as concise in supporting AP – could be enriched with minimal SAP features, experimental extensions to the Mastrave core exist to create a common basis for reasoning and encouraging mutual paradigm contamination. For example, transparent bindings for providing array semantic constraints within Python with NumPy (<http://numpy.scipy.org/>) and GNU R (<http://www.r-project.org/>) are under experimentation. Despite a GNU/Linux like environment is required and supporting free operating systems takes precedence over non-free platforms, Mastrave has successfully run within Cygwin (<http://www.cygwin.com/>) under Windows.

The author explicitly conceived the Mastrave library for supporting nontrivial data-transformation models which typically require the active involvement of experienced modellers. Nontrivial data transformations are usually subjected to scientific peer review as original contributes to environmental modelling. This subset of data-transformation models may easily be suitable and convenient to be reused as precious components of new environmental models. Their role may be to access the information of available datasets by aggregating, filtering, slicing collections of data and by composing multiple datasets for approximating missing information. However, understandability, expressiveness and sustainable maintainability (e.g. abstraction, ease of modification and innovation) of both these models and their libraries may be essential in deciding their long-term survival. Suggestively, Stroustrup [2005] highlights “that on the order of 200 new languages are developed each year and that about 200 languages become unsupported each year”.

Besides strictly limiting dependencies to well-established, stable and actively developed free software packages, documentation is also a priority. Mastrave knowledge management policy is to directly update thorough documentation within source codes in the form of semantically enhanced structured comments⁵ – part of a consistent set of coding standards. Online documentation (<http://mastrave.org/doc/>) only refers to stable modules – for which usually expert users provided feedback. It is automatically updated to persistently be in line with the most recent published version and for each function offers a permanent URL so that it can be safely cited within scientific publications. Examples of usage systematically highlight the extent of abstraction each module offers.

Interestingly, it might be far more difficult to practically implement several complex models rather than conceptually describe them: a subset of modelling generalizations may deal more with abstraction rather than with the explosion of lines of code. For such a subset of modelling applications, an approach in which “the advantages of executability and universality found in programming languages can be effectively combined, in a single coherent language, with the advantages offered by mathematical notation” [Iverson 1980] might be interesting.

2 SEMANTIC ARRAY PROGRAMMING

Array programming originated [Iverson 1980] to reduce the gap between mathematical notation and algorithm implementations by promoting arrays (vectors, matrices, tensors) as atomic quantities with extremely concise manipulating operators.

Coherent array-based mathematical description of models can simplify complex algorithm prototyping while moving mathematical reasoning directly into the source code – because of its substantial size reduction – where the mathematical description is actually expressed in a completely formalized and reproducible way. Two additional design concepts define semantic array programming [de Rigo 2012] as supported by the Mastrave library:

1. *modularizing* sub-models and autonomous tasks with a strong effort toward their *most concise generalization* and reusability in other contexts.
2. *semantically constraining* with array oriented –thus concise– invariants the information entered in and returned by each module instead of relying on external assumptions.

Modularization also expects consistent self-documentation of code along with a uniform predictable convention for module interfaces (without directly interfering with the preferred module’s implementation). Semantic constraints contribute to enforce within each module autonomous internal consistency checks instead of assuming the correctness of input information (e.g. instead of relying on object-oriented “safe” data). This way, even the residual monolithic management of data structures for exchanging information between modules is reduced at a minimum degree, by only relying on quite general data primitives such as multi-dimensional numerical arrays or collections of generic elements (cell-arrays).

2.1 An Example: Semantic Array-based Manipulation of Time Series

Array-oriented semantic constraints show similarities with behavioural subtyping [Liskov and Wing 1999], however without intrinsic links with object oriented programming. An example will highlight the way a SAP module interacts with external information provided by the module input arguments. Within the MATLAB language⁶ (which is dynamically and weakly typed) the concepts of vector, matrix and multidimensional array can all be

5 Analysing more than 2000 free software projects using MATLAB language, 23% of all lines of code are comments, while the share is over 50% for Mastrave’s GNU Octave/MATLAB code (<https://www.ohloh.net/languages/matlab>, <https://www.ohloh.net/p/mastrave/analyses/latest>).

6 The description applies to both GNU Octave and MATLAB computing environments.

represented by the native type *double* – which can also represent complex numbers.

A variable of type *double* can be an arbitrary array $\in \mathbb{C}^{n_1} \times \dots \times \mathbb{C}^{n_m} . \forall n_1 \dots n_m$ where \mathbb{C} is the set of 64 bit floating point complex numbers extended with $\pm\infty$ and *nan* (i.e. “not a number” denoting undefined or missing data) partially following the standard IEEE 754 for floating-point arithmetic [IEEE 1985, 2008]. Array operators concisely apply to these variables as if they were atomic, almost completely removing the need for explicit loops. As arrays are considered as a basic type, array variables can legitimately – and silently... – be assigned with other arrays of heterogeneous size. Despite this abstraction offers a powerful and extremely concise notation, complex array-based algorithms and models still may consist of hundreds or thousands of code lines, for which good practices such as assertions and ad-hoc consistency checks may (and should) tend to proliferate. This however poses relevant issues on the sustainability and effectiveness of such intensely needed “DIY” ad-hoc solutions. An example will illustrate how array-based, concise and consistent semantic constraints may help to immerse concise pieces of AP code within a robust semantic network of array-concepts.

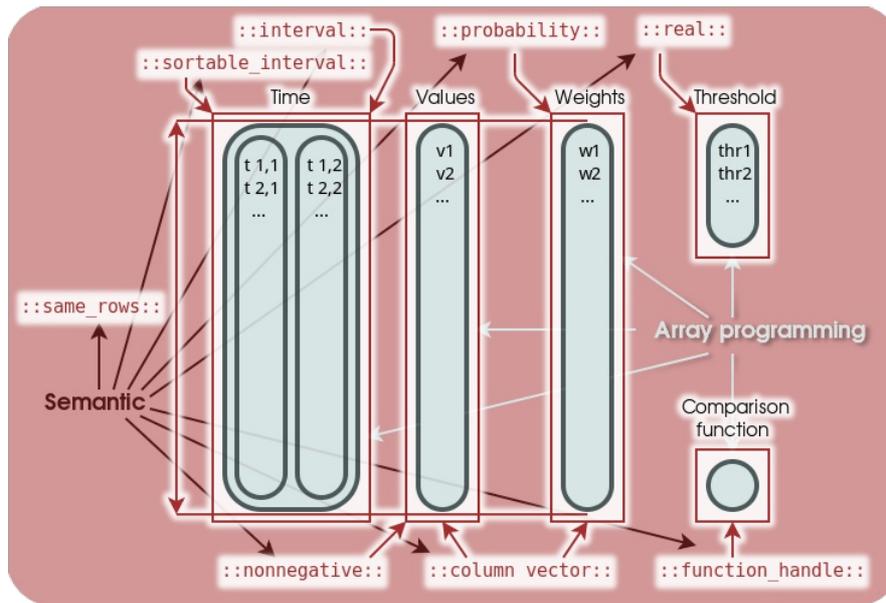


Figure 1. Array-oriented constraints determining (at run-time) the semantic structure of a time series, from the minimal preconditions (such as `::same_rows::`) up to submodel-specific requests (e.g. time interval sortability, values nonnegativity, ...).

Manipulating time series of data (Figure 1) is a frequent task to be addressed by specialized sub-models. For example, a rainfall time series may semantically be represented by a vertical set of time intervals, a corresponding set of rainfall values and a third sequence of weights – e.g. from 0 to 1 – providing an indication on the reliability of each value. Supposing the exemplified module needs to separately sort the time-series elements whose weights lie within a certain set of thresholds, then a collection of preconditions (each denoted as `::semantic-constraint::?`) should be satisfied:

1. the time series elements (time intervals, values and weights) must be composed by the same number of items (rows): `::same_rows::` ;
2. both values and weights arrays must be `::column_vector::S`;
3. values must be of real non-negative numbers (`::nonnegative::`) while weights must be in $[0, 1]$ (`::probability::`);
4. the time-interval array must be a two-column matrix: the second-column elements cannot be less than to the corresponding first column elements (`::interval::`);
5. intervals are required to be sortable, so that⁸ the intersection of any pair of intervals must be empty (`::sortable_interval::`);

7 http://mastrave.org/doc/mtv_m/check_is lists some 200 native semantic constraints.

8 The definition of intervals' sortability may be customized, depending on the specific task for which an ordering is required. However, an easy and unambiguous sufficient condition holds when all intervals are disjoint (`::sortable_interval::`).

6. the thresholds must be real numbers (`::real::`) and a custom comparison function is passed as optional module's input argument (`::function_handle::`).

Complexity of constraints' semantics varies from case to case and several constraints can only be tested at run-time. Constraints apply to vectors, matrices and often to multi-dimensional arrays, easing their concise application to complex data, so that nontrivial networks of semantic constraints (preconditions, invariants and postconditions) could be straightforward to describe. Unsatisfied constraints raise exceptions which can be managed to enable model self-healing, or simply can cause a human-readable and self-explaining error to abort the computation. The Mastrave module `check_is` implements constraints so to make them available as a rich, consistent and flexible set of semantic assertions systematically used and documented by all Mastrave modules.

Semantic array-programming filters may be used to *semantically wrap* pre-existing third-party models, since constraint errors can be easily exposed and used for strengthening third-party models' robustness. After this small example, the following section will introduce a wider SAP application to local-average invariant (LAI) downscaling of climate raster data.

3 APPLICATION TO LOCAL-AVERAGE INVARIANT DOWNSCALING

General circulation models are widely used for estimating patterns of temperature and precipitation under climate change scenarios. Their coarse spatial resolution is often unsuitable to be directly used within environmental models and can be mitigated by complementing large-scale information with that provided by regional climate models and local-scale data [Bates *et al.* 1998; Patz *et al.* 2005; Fowler *et al.* 2007].

Among the simplest downscaling methods, the *perturbation-method* [Prudhomme *et al.* 2002] or *delta-change, change-factor* method has been proposed for downscaling future climate projections at continental scale, up to a spatial resolution of 30 arc-second (less than 1 km at the Equator) [Ramirez and Jarvis 2010; Tabor and Williams 2010]. The method exploits the availability for the present (baseline) of information at a finer resolution than the one available for future climate scenario projections. Given a coarse-scale projection of climate *changes*, it is first upsampled by applying a smooth interpolation and then added to a higher resolution baseline. The result estimates the climate projection at the finer scale. Despite its simplicity, the change-factor method has been recommended for strategic assessment of multiple climate change scenarios [Wilby *et al.* 2004] and appears as well performing as more sophisticated methods in reproducing mean characteristics [Fowler *et al.* 2007].

An evident weakness of the method may lie in the application of over simplistic interpolation strategies which may degrade the already coarse original climate change information. For example, smoothing splines are often employed, probably also because of their out-of-box availability. For each coarse-resolution spatial cell, global or regional climate models generally provide a value representing the *average* of the modelled climatic quantity within that cell. However, downscaling interpolations are often performed by confusing the (known) *average* value of a given cell of the coarse grid to be upsampled with the corresponding (unknown) *punctual* value of the cell centroid⁹, which is required by the spline. As a consequence, if the downscaling data-transformation model were reversed and from the interpolated fine-resolution values the corresponding coarse-cell averages were re-computed, they *would not respect the original coarse-scale climate change values*.

Several strategies¹⁰ can be followed in order for the downscaling data-transformation modelling to locally preserve as average the values of the original raster data. A downscaling data-transformation fulfilling the constraint to be local-average invariant(LAI) is here denoted as LAI-downscaling. Non-LAI spline interpolation is natively supported by the Mastrave function `upsample`, which also provides LAI interpolations.

3.1 Importing Remote Data

Several datasets provide raster data tiled per geographical area or time interval

⁹ More properly, the "punctual" value is the one of the finer-grid cell in which the centroid of the coarse-grid cell lies. It is obviously *unknown* before the downscaling, and even if it were known, still the problem of imposing the local average invariance (LAI) of the smoothing spline across each coarse-grid cell would remain.

¹⁰ For example, iterative smoothing of residuals with usual splines with a final LAI interpolation to exactly impose the constraint; or deconvolution of a smoothed non LAI interpolation with high differentiability class. For details, see the `upsample` function documentation.

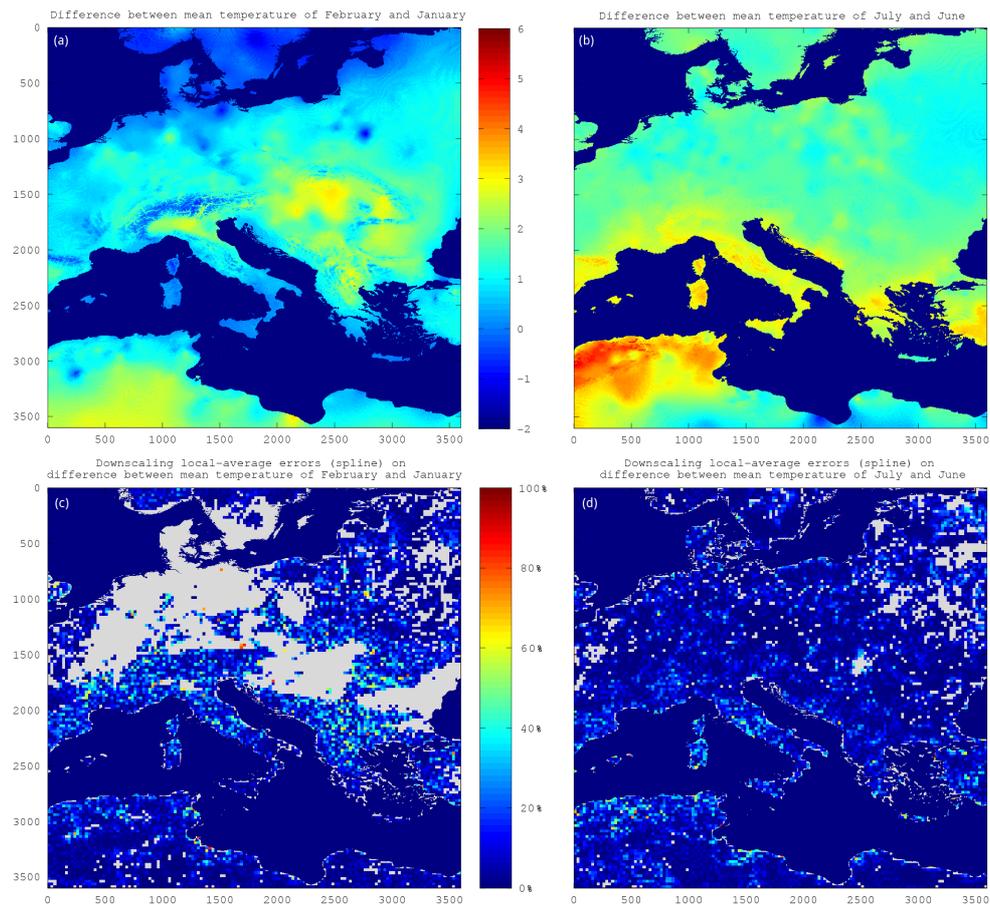


Figure 2. Benchmark difference (month-to-month) in °C/month between mean temperature [WorldClim 2012] of (a) February and January, and (b) July and June. The corresponding dimensionless ratio – (c) and (d) – between non-LAI interpolation local average errors (which using LAI approaches would be zero) and usual errors due to the difference between downscaled (via cubic splines) and “true” future values, is computed on the dense-grid and averaged in the coarse-grid. Grey pixels indicate where both errors are below the dataset uncertainty (± 0.05 °C). These errors are unnecessarily introduced using non-LAI smoothing: LAI-downscaling mathematically ensures the error-ratio maps to be null. Spatial units: 30 arc-seconds lat-lon with translated origin (top left).

[Bossard *et al.* 2000; Farr *et al.* 2007; WorldClim 2012]. Importing these data requires downloading, decompressing (if needed) and transforming projections and file formats to fit the needs of the computational modelling research. For example, intense array oriented manipulations of data benefit from accessing synchronously a plethora of large files in $O(1)$ time, read/write mode. These activities may imply manipulating a noticeable amount of original, temporary and working files. Mastrave supports importing remote data with a single command. Exemplifying with the WorldClim dataset, the command:

```
uri = @(files)[ 'http://www.worldclim.org/data/v1_4/tiles/cur/' ...
               files '_16_tif.zip'
               ]
mastrave( 'preload', uri( '{t{min,mean,max},alt,prec}' ) )
[filepath, filename, bytes] = fileinfo( '*.*bt' )
```

downloads the compressed archives of altitude, precipitation, temperature (monthly minimum, mean and maximum) of the WorldClim tile 16, extracts some 50 geotiff files and converts them in Binary Terrain format¹¹. The whole set of tiles might be accessed by replacing '*_16_tif.zip*' with '*_{0..4}{0..11}_tif.zip*' so that in this case almost 3000 geotiff raster data could be pre-loaded with a single command, by default enforcing a (customizable) strategy for avoiding replicated downloads. The `fileinfo` module eventually allows generating a list of available data while inspecting their size.

¹¹ <http://vterrain.org/Implementation/Formats/BT.html>. This file format is natively supported e.g. by the GDAL library [2012] and is straightforward to be accessed (read/write mode) in constant time.

3.2 Interpolation Errors of non-LAI Downscaling

Climate change involves complex relationships among seasonal and historic variability [Loarie *et al.* 2009]. The qualitative analysis of the interpolation errors due to the use of non LAI downscaling has been performed looking for a simplified benchmark example with monthly information, easy to reproduce and offering a reasonable numerical variety, despite completely decoupled from actual climate change projections. The month-to-month (m-m) differences between the WorldClim 1950-2000 mean temperatures (in North Africa and part of the Europe) of July and June, and of February and January, are shown in Figure 2.

They have been used as benchmark examples due to their complementary, almost opposite patterns of changes which could allow to qualitatively simulate the numerical effects of downscaling a wide range of possible climate changes. The average intensities of the two benchmark temperature changes are the two m-m annual local minima with positive value, and lie below almost all the projected temperature anomalies between 2000 and 2100. For upsampling (with both usual non-LAI cubic splines and a LAI interpolator), the Mastrave function `upsample` has been used. The function allows processing with a single invocation a series of raster-data even if they are multi-dimensional arrays. 30 arc-seconds is the original WorldClim resolution, preserved for the benchmark baseline. In order for the benchmark to simulate a coarser change-factor grid¹², the mean of all non-missing data within blocks of 25 x 25 cells is simply computed with the Mastrave function `mb1k_fun`. It enables separately processing rectangular blocks of arbitrary size (here 25 x 25 pixels) of a given raster grid, with arbitrary functions (here the mean) passed as input argument:

```
mb1k_fun( raster_data, @(x)mean(x(~isnan(x))), 25 )
```

whose meaning is straightforward when knowing that `~` is the logical negation and `@(x)...` is an anonymous function with `x` as input argument (“mean of `x` without its `nan` values”).

The ratio $\epsilon^{\text{non-LAI}}/\epsilon^{\text{down}}$, (dimensionless) between non-LAI interpolation errors $\epsilon^{\text{non-LAI}}$ (absolute difference between each change-factor’s coarse-grid pixel and the corresponding average of 25 x 25 non-LAI interpolated pixels of the dense-grid) and the 25 x 25 average of the absolute dense-grid errors between the interpolated and the “true” future values, ϵ^{down} , is shown in Figure 2 (c,d). Using LAI interpolation, $\epsilon^{\text{non-LAI}}$ would have been mathematically imposed to be null, so to also force (c,d) to be zero. This highlights the non negligible errors which non-LAI smoothing unnecessarily introduces in addition to the intrinsic downscaling error ϵ^{down} . While estimating ϵ^{down} is generally impossible in non-benchmark applications of the change-factor method, $\epsilon^{\text{non-LAI}}$ is always computable.

Computing less trivial statistics such as the velocity of the temperature change [Loarie *et al.* 2009] (km month⁻¹) with the benchmark data, would just need a few additional lines of code:

```
check_is( {T_Jan T_Feb lat lon}, 'same_size' )% semantic check: minimal example
noise    = rand( size( T_Jan ) )*0.1-0.05;    % WorldClim T uncertainty: 0.1 °C
k        = [1;2;1]*[-1;1]                    % slope kernel
[dx,dy]  = deal( mfilter( {k, k'}, T_Jan + noise, [] ){:} );
lat2km   = lat * 111.325                    % conversion latitude-degree → km
lon2km   = cos( pi/180 * lon ) * 111.325    % conversion longit.-degree → km
T_speed  = ( T_Feb - T_Jan ) ./ hypot( dx.*lat2km, dy.*lon2km )
```

where `T_Jan`, `T_Feb`, `lat`, `lon` are the mean WorldClim temperatures of January and February, and the latitude and longitude raster data in degrees. The velocity of change is defined as the ratio between the temporal gradient (here `T_Feb - T_Jan` °C month⁻¹) and the spatial gradient (the module of the vector with components `dx` and `dy`, °C km⁻¹). Each pixel is added with uniformly distributed random noise in ± 0.05 °C for mitigating – within the dataset uncertainty which for WorldClim temperature is 0.1 °C – “the incidence of flat spatial gradients that cause infinite speeds”, as suggested by Loarie *et al.* [2009]. Comparing this code (also the Mastrave `mfilter`) with the mathematical description in [Loarie *et al.* 2009] easily shows that conciseness could also enhance model understandability. Outside benchmark examples, computing the velocity of change for a given quantity between current and future scenarios could typically use coarser change-factors for estimating fine-scale future information: in this case, climate change upsampling should be addressed with LAI downscaling.

CONCLUSIONS

Semantic array programming is proposed as a powerful conceptual and practical tool

¹² This is done while still preserving the possibility of comparing modelled results with the “true” future denser grid: which, outside this benchmark simulation, would typically be impossible.

(supported by the Mastrave library) for easing scalability and semantic integration in environmental modelling, by complementing extremely concise manipulating operators with modularization and compact semantic constraints. Mastrave usage has been exemplified with a regional scale benchmark application to local-average invariant (LAI) downscaling of climate raster data, whose straightforward use in Mastrave could help to promote LAI downscaling as a more correct approach for upsampling grids of climatic spatial averages.

Revisiting a classic framework for Information Technology benefits [Maggiolini, 2011], semantic array programming conciseness – as implemented in Mastrave – may positively affect several environmental-modelling aspects: from reducing new models' *production* costs (because of its reusable modules and the drastic decrease of code lines inherent to adopting the AP paradigm), to mitigating *coordination* costs among data and sub-models (because of coherent, compact abstraction and the ability to transparently import large remote data) finally also enabling *communication* economies (achieved through extremely compact, reliably tested semantics, also suitable to *semantically wrap* legacy models).

ACKNOWLEDGMENTS

Mastrave is hosted by the GNU Savannah¹³ free software forge and reviewed by the Free Software Directory¹⁴: the code review required by Savannah improved the library structure and licensing. Semantic array programming was conceived by also considering the epistemological implications¹⁵ of free software [Stallman 2009]. Many students, computational and environmental modelling scientists provided valuable user's feedbacks along with the Maieutike Research Initiative volunteers. A metadata analysis on semantic data-transformation models is actively investigated with G. Guariso, while research is ongoing on applications of LAI downscaling with C. Bosco, G. Caudullo and J. I. Barredo.

REFERENCES

- Angelis-Dimakis, A., M. Biberacher, J. Dominguez, G. Fiorese, S. Gadocha, E. Gnansounou, E., G. Guariso, *et al.*, [Methods and tools to evaluate the availability of renewable energy sources](#), *Renewable and Sustainable Energy Reviews*, 15(2), 1182–1200, 2011.
- Bates, B.C., S.P. Charles, and J.P. Hughes, [Stochastic downscaling of numerical climate model simulations](#), *Environmental Modelling & Software*, 13(3–4), 325–331, 1998.
- Bosco C., D. de Rigo, J. Poesen, O. Dewitte and P. Panagos, Modelling soil erosion at European scale. Harmonisation and reproducibility. (in prep.).
- Bossard, M., J. Feranec, and J. Otahel, [CORINE land cover technical guide - Addendum 2000](#), *Technical report No 40*, European Environment Agency, 2000.
- Casagrandi, R., and G. Guariso, [Impact of ICT in Environmental Sciences: A citation analysis 1990-2007](#), *Environmental Modelling & Software*, 24(7), 865–871, 2009.
- Castelletti, A., D. de Rigo, L. Tepsich, R. Soncini-Sessa, and E. Weber, [On-Line Design of Water Reservoir Policies Based on Inflow Prediction](#), *Proceedings of the 17th IFAC World Congress*, 14540–14545, 2008.
- Crowley, T.J. and W.T. Hyde, [Transient nature of late Pleistocene climate variability](#), *Nature*, 465(7219), 226–230, 2008.
- de Rigo D. and C. Bosco, [Architecture of a Pan-European Framework for Integrated Soil Water Erosion Assessment](#). Environ. Software Systems. *Frameworks of eEnvironment, IFIP Advances in Information and Commun. Technology*, 359, Chapter 34, 2011.
- de Rigo D. and C. Bosco, Architecture of a framework for integrated natural resources modelling at regional scale: application to pan-European soil water erosion assessment. (in prep.).
- de Rigo, D., A. Castelletti, A.E. Rizzoli, R. Soncini-Sessa, and E. Weber, [A selective improvement technique for fastening Neuro-Dynamic Programming in Water Resources Network Management](#). *Proceedings of the 16th IFAC World Congress*, 7–12, 2005.
- de Rigo D., G. Caudullo, G. Amatulli, P. Strobl and J. San-Miguel-Ayanz, Modelling tree species distribution in Europe with constrained spatial multi-frequency analysis (in prep.).
- de Rigo, D., A.E. Rizzoli, R. Soncini-Sessa, E. Weber, and P. Zenesi, [Neuro-dynamic programming for the efficient management of reservoir networks](#). *Proc. of MODSIM 2001, International Congress on Modelling and Simulation*, 4, 1949-1954, 2001.
- de Rigo, D. [Semantic Array Programming with Mastrave - Introduction to Semantic Computational Modelling](#). *The Mastrave project*, to appear in 2012.

13 <http://savannah.nongnu.org/projects/mastrave>

14 <http://directory.fsf.org/wiki/Mastrave>

15 <http://mastrave.org/research.html> provides information on volunteering within research collaborations and on expected scientific code review process and knowledge freedom.

- Eaton, J.W., D. Bateman, and S. Hauberg, [GNU Octave Manual Version 3](#). Network Theory Limited, 2008.
- Edwards, J.L., [Research and Societal Benefits of the Global Biodiversity Information Facility](#). *BioScience*, 54(6), 486–487, 2004.
- Estreguil, C., D. de Rigo, and G. Caudullo, Towards a standardised characterisation of habitat pattern. (to be submitted to *Environmental Modelling & Software*).
- Estreguil, C., G. Caudullo, D. de Rigo, C. Whitmore, and J. San-Miguel-Ayanz, Reporting on European forest fragmentation: Standardised indices and web map services. (submitted to *Earthzine, IEEE*).
- Farr, T., P. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, et al., [The Shuttle Radar Topography Mission](#). *Reviews of Geophysics*, 45 RG2004, 2007.
- Fowler, H.J., S. Blenkinsop, and C. Tebaldi, [Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling](#), *International Journal of Climatology*, 27(12), 1547–1578, 2007.
- Geospatial Data Abstraction Library, version 1.8, *Open Source Geospatial Foundation*, <http://gdal.osgeo.org> (last accessed 2012-02-26).
- Hashimoto, S., Morishita, T., Sakata, T., Ishizuka, S., [Increasing trends of soil greenhouse gas fluxes in Japanese forests from 1980 to 2009](#), *Scientific Reports*, 1, n.116, 2011.
- Haughton, A.J., Bond, A.J., Lovett, A.A., Dockerty, T., Sünnerberg, G., Clark, S.J., Bohan, et al., [A novel, integrated approach to assessing social, economic and environmental implications of changing rural land-use: a case study of perennial biomass crops](#). *J. of Applied Ecology*, 46(2), 315–322, 2009.
- Haylock, M.R., N. Hofstra, A.M.G. Klein Tank, E.J. Klok, P.D. Jones, and M. New., [A European daily high-resolution gridded dataset of surface temperature and precipitation](#). *J. Geophys. Res (Atmospheres)*, 113, D20119, 2008.
- Hijmans, R.J., S.E. Cameron, J.L. Parra, P.G. Jones, and A. Jarvis, [Very high resolution interpolated climate surfaces for global land areas](#), *International Journal of Climatology*, 25(15), 1965–1978, 2005.
- IEEE, [IEEE standard for binary floating-point arithmetic](#). Std. 754-1985, 1985.
- IEEE, [IEEE standard for floating-point arithmetic](#). Std. 754-2008, 2008.
- Iverson, K. E., [Notation as a tool of thought](#). *Commun. of the ACM*, 23, 444-465, 1980.
- Liskov, B.H. and J.M. Wing, Behavioral Subtyping Using Invariants and Constraints, *MU CS-99-156*, School of Computer Science, Carnegie Mellon University, 1999.
- Loarie, S.R., P.B. Duffy, H. Hamilton, G.P. Asner, C.B. Field, and D.D. Ackerly, [The velocity of climate change](#), *Nature*, 462(7276), 1052–1055, 2009.
- Lovett, A., G. Sünnerberg, G. Richter, A. Dailey, A. Riche, and A. Karp, [Land Use Implications of Increased Biomass Production Identified by GIS-Based Suitability and Yield Mapping for Miscanthus in England](#), *BioEnergy Research*, 2(1), 17–28, 2009.
- Maggiolini, P., [Information Technology Benefits: A Framework](#). Emerging Themes in Information Systems and Organization Studies, 281-292, 2011.
- Merritt, W.S., R.A. Letcher, and A.J. Jakeman, [A review of erosion and sediment transport models](#), *Environmental Modelling and Software* 18(8-9) 761–799, 2003.
- Patz, J.A., D. Campbell-Lendrum, T. Holloway, and J.A. Foley, [Impact of regional climate change on human health](#), *Nature* 438(7066), 310-317, 2005.
- Prudhomme C., N. Reynard, and S. Crooks, [Downscaling of global climate models for flood frequency analysis: Where are we now?](#), *Hydrological Processes* 16: 1137–1150, 2002.
- Ramey, C., and B. Fox, [The GNU Bash Reference Manual](#). Network Theory Limited, 2006.
- Ramirez, J., and A. Jarvis, [Downscaling Global Circulation Model Outputs: The Delta Method](#), *Decision and Policy Analysis Working Paper No. 1*, CIAT, 2010.
- Scholes, R.J., G.M. Mace, W. Turner, G.N. Geller, N. Jürgens, A. Larigauderie, et al., [Toward a Global Biodiversity Observing System](#), *Science*, 321(5892), 1044-1045, 2008.
- Stallman, R.: [Viewpoint: Why "open source" misses the point of free software](#). *Commun. ACM*, 52(6), 31–33, 2009.
- Stroustrup, B. [A rationale for semantically enhanced library languages](#). *ACM LCSD05*, 2005
- Tabor, K. and Williams, J.W., [Globally downscaled climate projections for assessing the conservation impacts of climate change](#), *Ecological Applications*, 20(2), 554–565, 2010.
- Trivedi, M.R., P.M. Berry, M.D. Morecroft, and T.P. Dawson, [Spatial scale affects bioclimate model projections of climate change impacts on mountain plants](#), *Global Change Biology*, 14(5), 1089–1103, 2008.
- Yesson, C., P.W. Brewer, T. Sutton, N. Caithness, J.S. Pahwa, M. Burgess, W.A. Gray, et al., [How Global Is the Global Biodiversity Information Facility?](#) *PLoS ONE* 2(11): e1124, 2007.
- van den Besselaar, E.J.M., M.R. Haylock, G. van der Schrier, and A.M.G. Klein Tank, [A European Daily High-resolution Observational Gridded Data set of Sea Level Pressure](#), *J. Geophys. Res.*, 116, D11110, 2011.
- Wilby R.L., S.P. Charles, E. Zorita, B. Timbal, P. Whetton, and L.O. Mearns, [Guidelines for use of climate scenarios developed from statistical downscaling methods](#), *Supporting material of the Intergov. Panel on Climate Change*, DDC of IPCC TGCIA, 27, 2004.
- WorldClim, Data Format, <http://www.worldclim.org/formats>, (last accessed 2012-02-26).