Contact at Harris

Ing. Federica Selva, 039 6058605 int 18,

[federica.selva@harris.com](mailto:federica.selva@harris.com)

Stefano Gagliano

335 7611436

[sgaglian@harris.com](mailto:sgaglian@harris.com)

## The problem

Constrained minimization

A number of parameters of a dynamic crop growth are optimized against observations (remote sensing, eddy flux data). The model is dynamic in the sense that it uses as input a number of meterological time series, so no look-up table approaches are possible.

## Current approach

Use of MPFIT procedures (constrained Levenberg-Marquardt optimization).

Pixel based approach (one run for each pixel, potentially millions of pixels; and one run for each season)

## Limitations to be improved:

1. The optimization can be trapped into local minima
2. Speed (1 minute for one season), not feasible to apply to image time series

## Type of contract

Procurement (less than 15000), relatively easy. In 2017, final invoice can be lower than offer.

## Organization of the code and input/output directories

On my pc I have a workspace with several projects, some of them linked. To share the code with you I have copied all the relevant ones under a single workspace, “simmod for Harris”. Please note that some projects contain a lot of routines, only few used by the algorithm under investigation. “Couple Model” is the project to be built.

## Brief code description of SimMod

A crop growth model (simForwMod\_water\_lim) describes the photosynthesis (GPP) of a generic crop and, linked to this, its biomass amount and its spectral reflectance (and NDVI as it would be observed by MODIS sensor).

**The GPP and NDVI are simulated at daily time step using meteorological inputs (ECMWF) and a number of model parameters.**

In the “inversion process” these parameters are “tuned” by minimizing the cost function defined on the difference between modelled and observed NDVI.

The performances of the model are checked by comparing the simulated GPP against a measured one (referred to as “eddy flux” GPP throughout the code).

The algorithm is launched running **A\_simInvMod\_handler\_water\_lim.pro** that reads an ini file to set the required parameters. Note that the full path of the ini file is hard coded in A\_simInvMod\_handler\_water\_lim.pro (line 33). Note that the full path to the output directory and the site\_ini filename must be given. This site\_ini contain the full path to the meteo, MODIS and eddy files used. For the time being, the code does not use actual images but temporal profiles extracted from actual images (the input files in csv format listed in site\_ini.csv).

Once the data are loaded the code first call another big chunk of IDL code (through **bridge\_SimMod2Phenot365.pro**) that should be neglected here. This is a parallel algorithm, involving optimization as well, that compute the phenology (start and end of season) from the NDVI time series. Just ignore it, it is used here to have a first guess of the timing of the start of season, used by the algorithm of interest.

The background reflectance is computed using the full MODIS time series (looking at minima of NDVI) using **rback\_from\_MODIS\_TS.pro**. Here another inversion is done to fit the Price function to the retrieved reflectance. Also this inversion should not be considered, probably fast enough and executed once (per site).

Then the time-axis of all variables is transformed to Julian days. The time domain is then set using SOS and EOS from the pheno-retrieval first guess (enlarged by nSD standard deviations). The season must be found there by the model. The time series of all variables are extracted for this time domain only.

First guess and constraints are set. Note that the only free variable of the radiative transfer module (PROSAIL) is the SLA.

If the parameters of the water limitation part are fixed, bewm (backwards exponentially weighted mean), bewm\_et0 and their ratio (used as stress factor) is calculate once and for all and stored in a common block.

Then it comes the slow part of the code involving the optimization is called (**InvWrapper\_water\_lim.pro**).

So, moving to **InvWrapper\_water\_lim.pro**, here I set the required info for the inversion made using MPFIT. Please ignore “Metropolis” inversion that was just an attempt. After that MPFIT is called. Please note that I adapted the standard MPFIT optimization to something I called upper envelope optimization, basically an interaction of MPFIT to be closer to the “upper envelope” of observed NDVI. Easier to talk about if needed.

MPFIT basically calls **MPFIT\_NDVI\_from\_model\_water\_lim.pro** again and again to evaluate the cost function (modelled NDVI – simulated NDVI) and compute gradient. In turn, MPFIT\_NDVI\_from\_model\_water\_lim.pro calls the crop growth model **simForwMod\_water\_lim.pro** with the proper input data. NOTE that the evaluation of the function is dynamic because one of the parameter is the start of the season, which in turn determine the temporal window to extract the input data. So the model parameters DO have effect on the input data. That’s why I could not implement a Look Up Table approach for the inversion.

|  |
| --- |
| **Notes on time in the inversion**  **1.**  When the temporal domain is set by pheno, it is a matter of simulating NDVI with the forward model that better fit the observed NDVI. This is made changing the model parameters and looking for the best fit. In the iterative optimization all the free parameters are treated equally despite there is a substantial difference between DOY0 (the time at which the growth starts) and all other parameters.  DOY0 determines the time at which the growth happens, therefore with which input “climate” the forward model will work.  The other parameters instead control how quickly (slowly) the vegetation will grow and decay given the input climate the forward model have.  Model parameters interact with climate (determined by the time window, in turn determined by DOY0), therefore, to better fit NDVI(time), I will have different parameters depending on the time considered.  But, as said, all parameters are treated equally, so it is likely that MPFIT cannot explore properly the space of DOY0 + other parameters (as in principle each single DOY0 open a scenario of possible parameters).  For this reason Dominique suggested to first slice the time using a proper step (8 days?) and the run the inversion with DOY0 fixed on each slice. Then get the best fitting and run it again ther with DOY0 free but constrained between the nearest slices. Looks a good idea, but will be slower than current one..  2.  The use of DOY0 as a standard inversion parameters poses an additional problem. DOY0 is discrete whereas inversion parameters are typically floating. So MPFIT will call the forward modal with floating DOY0. This is managed as follows. The forwards model is called with FLOOR(DOY0) first and the with CEIL(DOY0). Than the NDVI returned is the weighted average of the two simulations. Weighting being the decimal fraction of DOY0.  NOT THAT **THIS MAKES THE CODE SLOWER** BECAUSE THE FORWARD MODEL IS CALLED TWICE. |

Once the parameters are optimized the code goes back to main (A\_simInvMod\_handler\_water\_lim.pro) that print the results.

Final remark: data are stored in ‘E:\SimMod\_data’ on my pc, and are provide you in the folder ‘SimMod\_data’ under ‘\simmod for Harris’. You’ll have to change this reference (line 33 of A\_simInvMod\_handler\_water\_lim.pro, and references in the ini files ‘SimMod\_all\_ini2rsoilPrice.txt’ nad site\_ini.csv).

Once this is done and the code launched you’ll see results of 3 years for the site called SDDEM, and 5 years of the site called ESLMA, for a total of 8 inversions.