Report on stay at ZAMG

12/06/2017 – 21/07/2017

Implementation of Stochastic Pattern Generator (SPG) in ALADIN code

Stochastically Perturbed Parameterized Tendencies (SPPT) has been the subject of several LACE stays in the previous years. The global version of the scheme was successfully used by ECMWF (Buizza et al., 1999) and tuned with a spectral pattern generator during a main revision (Palmer et al., 2009). The limited area version and AROME extension was implemented by Francois Bouttier from Météo France (Bouttier et al., 2012). The detailed examination and further extension to ALARO took place in a framework of a LACE stay, 2014 (Szűcs, 2014). In the following year some possible developments were investigated and tested as problems were also reported (Szűcs, 2015). One of these problems is the unsatisfying behave of the random pattern generator in LAM version which made a motivation for its revision and start of the implementation of a new Stochastic Pattern Generator (SPG, Tsyrulnikov and Gayfulin, 2016) which was in the focus of a LACE stay last year (Szűcs, 2016a). Some details about such investigations and work was represented also on ALADIN-HIRLAM (Szűcs, 2016b), SRNWP-EPS (Szűcs, 2016c) and HIRLAM-EPS (Szűcs, 2016d) workshops.

Author:
Mihály Szűcs (OMSZ)
szucs.m@met.hu

Supervisor:
Martin Belluš (SHMU)
martin.bellus@shmu.sk

Christoph Wittmann (ZAMG)
christoph.wittmann@zamg.ac.at

1. Problematic issues with the current spectral pattern generator

The problems of the current random pattern generator was well-detailed in the first part of my previous report (Szűcs, 2016a). Now only the main points are being highlighted to give a motivation for the implementation of the new Stochastic Pattern Generator (SPG).
Theoretically the main disadvantage of the current pattern generator is that the same time correlation belongs to all the spatial scales. While forecast error is connected to atmospheric motions it would be beneficial to have various scales with separated spatial and time correlations for error representation purpose. With the current pattern generator the only possible way is to meet this goal if more random patterns are defined and applied during the same time (Palmer et al., 2009). This kind of solution can handle more scales on a discrete way but can not represent the continuous spectra of the motions.
In practice the LAM version of the current pattern generator can not work how it supposed to be. Its standard deviation should be controlled from the namelist but in practice it is much more than it is set. At the same time horizontal correlation is much smaller than its namelist-defined value and additionally this difference is domain-size dependent.

2. Description of a Limited-Area Spatio-Temporal Stochastic Pattern Generator

In the previous part the theoretical and practical deficiencies of the current pattern generator have been underlined. So the two main requirements for a new one are the following:
- For representing the model error belonging to various scales there should be different time correlation values connected to different spatial correlation values. This feature is called as the “proportionality of scales”.
- The new pattern generator should be correctly tunable: the namelist-defined values should be identical (or at least close enough) to the statistical values calculated from the generated fields.
The theoretical background of the SPG scheme is well-described by the inventors in an article (Tsyrulnikov and Gayfulin, 2016) and in a document attached to the external code of SPG. This FORTRAN code can be freely downloaded from the github with additional technical documentation:
https://github.com/cyrulnic/SPG
In this report the focus is not on the theoretical details but on the technical implementation and on properties which can effect the results and their usage. To meet this purpose just some interesting features are outlined:
- How it was already mentioned, the spatio-temporal covariances should obey the “proportionality of scales” principle: larger (shorter) spatial scales should be associated with larger (shorter) temporal scales.
- The SPG should produce univariate stationary in time and homogeneous and isotropic in space Gaussian pseudo-random fields. Note that also the authors are interested in non-Gaussian noise which can fit better to some meteorological variables.
- External SPG code is able to produce 2D and 3D random fields, as well.
- The current version of external SPG code works with 3rd order in time spectral-space based solver. It makes it easier to implement in ALADIN code where the current pattern generator is applied also in spectral-space. Note that authors have interest in physical-space solvers, as well.
- The external SPG code is tuned with some changes which can save a large amount of computational cost without significantly effecting the statistical behave of the fields.

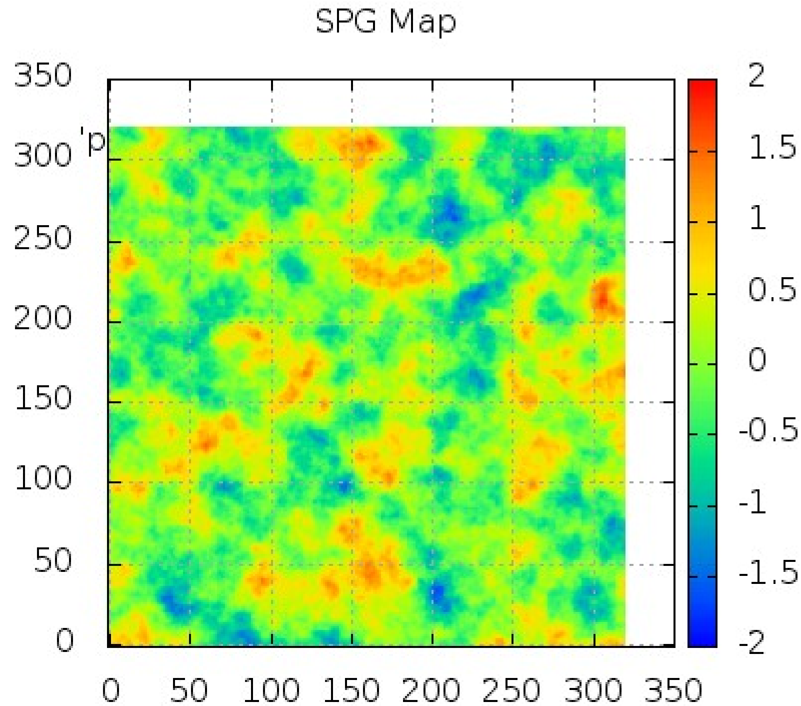

Fig.1: Random field generated by SPG (as an external program).

This program can be run as an external one with a configuration file where model dependent parameters (e.g. model size and timestep) can be defined in advance. Some additional printout line can help visualize the pattern generated by this program (Fig.7.).

The main steps of the algorithm can be highlighted as the following:
- Read and initialization of settings
- Pattern generation
    - Initialization of some additional variables
    - Loop on the different wavenumbers (in 2D or 3D) and calling the solver separately
        - Calling for Gaussian noise
        - Loop on the different eps members (samples)
            - Initialization of the random numbers of SPG at time=0 using the Gaussian noise
        - Calling for Gaussian noise
        - Loop on the different eps members
            - Loop on the different SPG-timesteps
                - Evolving the random numbers in time using Gaussian noise
    - Fast Fourier Transformation (FFT)
- Calculation of gridpoint statistics

After the examination of these points it became clear that a quite big part of the code is responsible for the Gaussian noise generation and for the FFT. While such algorithms are also available in ALADIN code it became clear that the rest of it is easier to implement than generate thousands of fields with an external program and than read and use during a model integration.

3. Implementation of the SPG into the ALADIN code

As it was mentioned above there is no need to implement the whole SPG program into the ALADIN code. However there are two main related questions:

- Where can it be implemented?
- How can it be reorganized?

At the first test we would like to use the random fields of the SPG on the same way in the SPPT than it is with the current random pattern generator. So the easiest way was to implement everything at the same part of the model where the current pattern generator works. Technically speaking it means that an additional switch can enable the SPG method in the initialization (suspsdt routine) and in the calculations in spectral space (functions of spectral_arp_mod module file). These calculations are called from the very-high-level stepo routine. The above-mentioned switches can make it easy to the user to decide if the current pattern generator or SPG would be applied.

A very tricky part of the external code implementation is the initialization. To set the time and spatial correlation length in the external program two config parameters are to responsible for. L05 and T05 are showing respectively the value where and when the spatial and time correlation functions are 0.5. Via an iterative process λ and μ values are calculated which are actually used in SPG equations. As a part of this iterative process FFT calculations are called several times. Normally in ALADIN code FFT and IFFT are called during the integration when we switch from spectral to physical-space or back. It did not seem straightforward to use transformation immediately in the setup routine so in practice the following way was used: First the L05 and T05 are defined in the external program together with the domain and timestep information and than it calculates the correct λ and μ values. This step is necessary only once for a given model configuration and for a given L05 and T05 setting. After that the evaluated λ and μ can be set as a namelist parameter of ALADIN implementation. This is not a really nice solution but needs only short time at the beginning of a test.

The structure of the pattern generation needed a massive reorganization because of the order of the loops. Of course in our case all the eps members are independent model runs so their loop has to come on the highest level. What is even more interesting that the loop on the wavenumbers and on the timesteps has to be switched.

An extra problem is that the SPG works with a different timesteps than the NWP model itself. Additionally this timestep is wavenumber dependent and also effected by the tuning which makes the code faster. To handle this problem for every wavenumber there is an extra calculation which defines a number of substeps and their length which is used by the SPG. How it was already mentioned the solver is 3^{rd} order in time which means that for calculation of a new value, we need the value of the previous three substeps. That means that the storage of the last three substep fields had to be also handled because they are needed to evolve values over model timesteps. We can also note that independent Gaussian-noise is necessary for every substeps which means that vector (which is filled by these random values at the beginning of the model timestep) size has to be increased in accordance with the wavenumber dependent substep number.

The schematic structure of SPG in ALADIN can be described as the following:

- 1st call of the generator from the suspsdt subroutine
    - Initialization of some additional variables
    - Calling for Gaussian noises (same way as in current pattern generator, but more times)
    - Loop on the different wavenumbers
        - Calling for the solver (this is a new function in spectral_arp_mod)
            - Initialize the SPG random number for the given wavenumber (just in 2D)
            - Loop on the substeps
                - Evolve SPG random number of the given wavenumber
            - Store the last three substeps of the scheme
- Fast Fourier Transformation (same way as in current pattern generator)
- Using grid-point values to perturb total tendencies (same way as with the current pattern generator)
- Loop on the model timesteps
    - Further call of the generator from the stepo subroutine
        - Calling for Gaussian noises (same way as in current pattern generator, but more times)
        - Loop on the different wavenumbers (just in 2D)
            - Calling for the solver (this is a new function in spectral_arp_mod)
                - Restore the SPG random number of the last 3 substeps
                - Loop on the substeps
                    - Evolve SPG random number of the given wavenumber
                - Store the last three substeps of the scheme
- Fast Fourier Transformation (same way as in current pattern generator)
- Using grid-point values to perturb total tendencies (same way as in current pattern generator)

An additional challenge was that external SPG program works with rectangular truncation while ALADIN uses elliptic truncation. It needed a careful revision on the total wavenumbers, as well.
It has to be noted that in this implementation of the SPG only a 2D version became available. Doing this way was more simple and in accordance with the way how we (and also ECMWF) currently use random patterns. Of course, a possible direction of developments could be to implement the 3D version, as well.

The code implementation was done under cycle 38 and its pack (subroutines, binary) is available on cca:
/perm/ms/hu/hu7/pack/38t1_new_spg.01.MPI631INTEL150.x
As the modified routines are usually not really touched cycle by cycle it looks easy to phase the implementation into cycle 40.

4. Fields and their statistical behave

First it can be demonstrated that fields are good-looking and qualitatively better than the ones with the current pattern generator (see previous reports). That means that there are no strange spots filled by -1 or +1 values, the spatial and time structures look reasonable and the “proportionality of scales” feature is visible. Additionally such figures can be compared with the ones which are in the SPG inventor's publications and documentations.
To get the following results Hungarian AROME domain was used. For standard deviation σ=0.5 was applied while L05=100km and T05=1hour. Many results are shared via google documents which is useful in case of gif animations:
https://drive.google.com/open?id=0B87pwSTd4-_mTGNPN0xfMkpkVzA

Fig. 2. shows a status of a random pattern at a given timepstep. Its time evolution can be followed on a gif animation at the above-mentioned webplace.

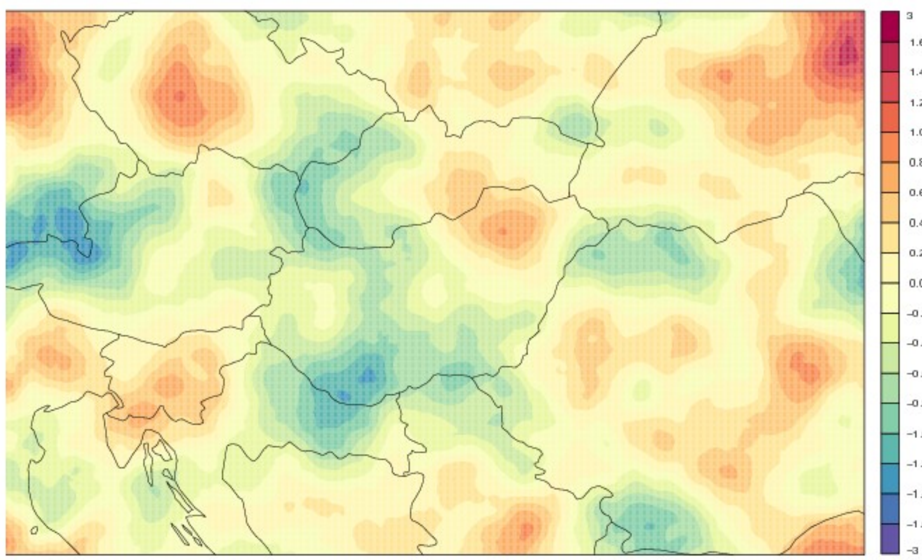

Fig.2: Random field generated by SPG (in ALADIN code implementation) for Hungarian AROME domain.

If we visualize the x-oriented cross-section of the values the small scale structures can be even better recognizable which is an advantageous feature of SPT in comparison with the current pattern generator (Fig.3.). If the timeseries of such pictures is visualized as a gif animation even their faster change is racognizable (see at the google link).
It is also possible to pick just one point of the field and visualize its time evolution (Fig.4).

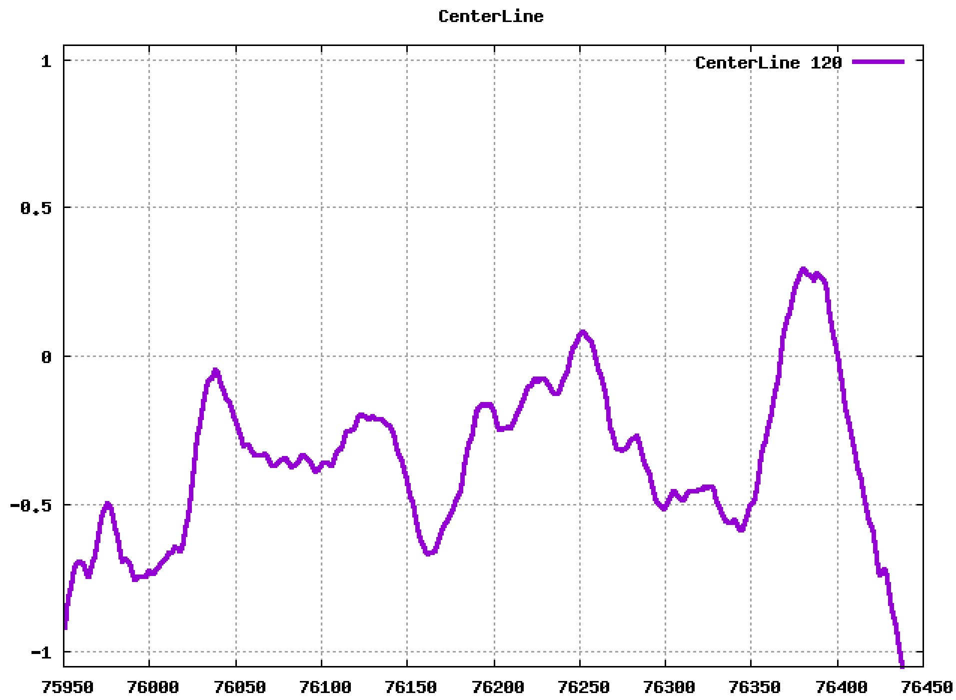

Fig.3: An x-oriented cross-section of the random pattern generated by SPG (in ALADIN code implementation).

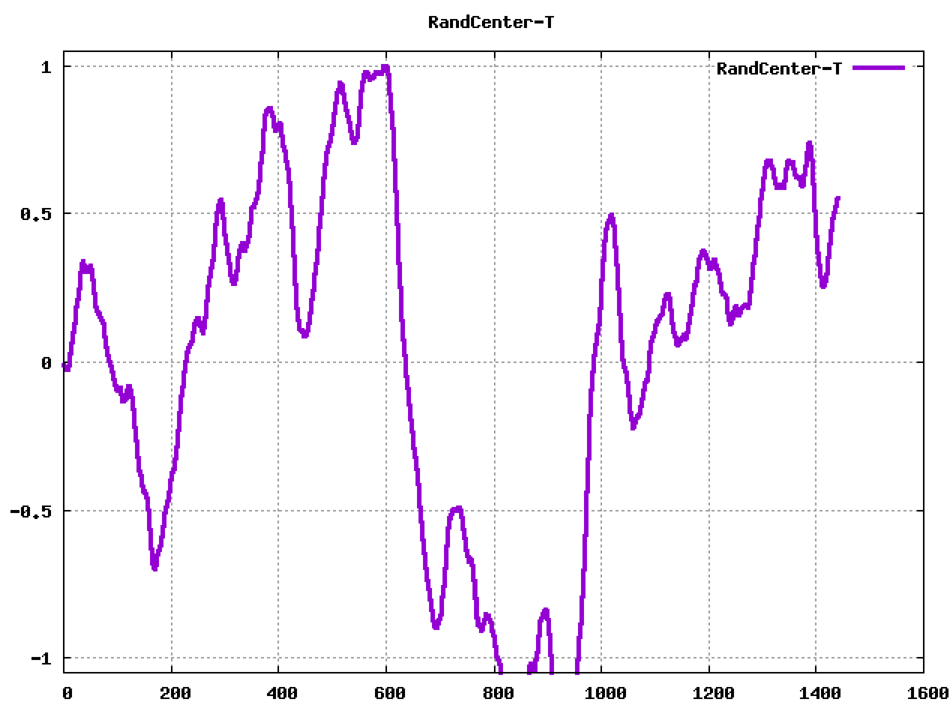

Fig.4: The time evolution of the random value of a given gridpoint in the center of the domain.

Of course such a qualitative comparison can not be absolutely satisfactory. The statistical behave of the pattern was investigated over 10 runs which length was set to 6hours.
The standard deviation calculated over such a relatively big sample was 0.502 which is close enough to the namelist defined value. Note that with the current pattern generator it is around 1.2 if we omit clipping which can significantly decrease that at the end.
How it was mentioned in previous reports the histogram of random numbers did not have a Gaussian shape with the current pattern generator. With SPG it looks much better from this perspective (Fig.5.).

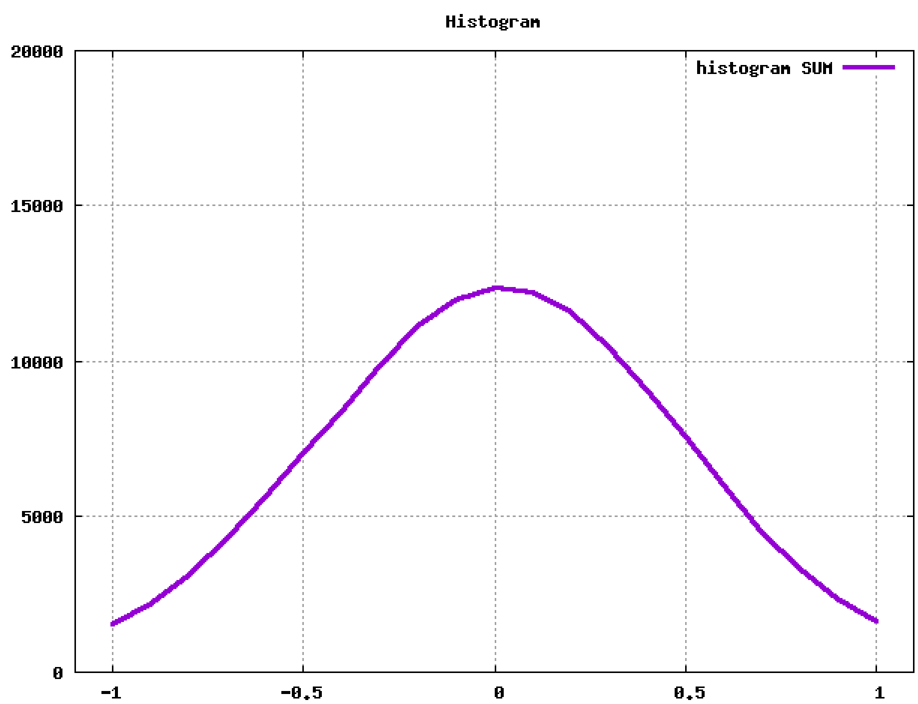

Fig.5.: The histogram of the random numbers averaged over SPG generated random fields.

The spatial and temporal correlation functions have been also checked. They give very important feedbacks about the correctness of the implementation. We can compare the functions calculated from the ALADIN code and from the external program evaluated fields. It is also possible to check if the decreasing correlation functions reach 0.5 value how it is set by L05 and T05 or not.
Fig.6. shows the spatial correlation functions and Fig.7. represents the temporal correlation functions. It is obvious that the lines belonging to ALADIN and to external program calculations are quite close.
ALADIN version crosses 0.5 level between 107.5 and 110km if function is calculated in direction of x and between 102.5 and 105 in direction of y. In my experience the these values are getting closer and closer to the theoretical value how sample size is increased so this difference looks acceptable. Note that in the current pattern generator we can get quite similar values if we set 4000km as horizontal correlation length for Hungarian AROME domain. In such case correlation functions are crossing 0.5 level between 110 and 112.5km in direction x and between 87.5 and 90km in direction y. So SPG looks also better from the aspect of isotropy.
If we examine temporal correlation function the situation looks even better. It reaches 0.5 between 59 and 60 minutes. Note that in the current pattern generator this value is around 4 hours if we set 6 hours as decorrelation time length in the namelist.

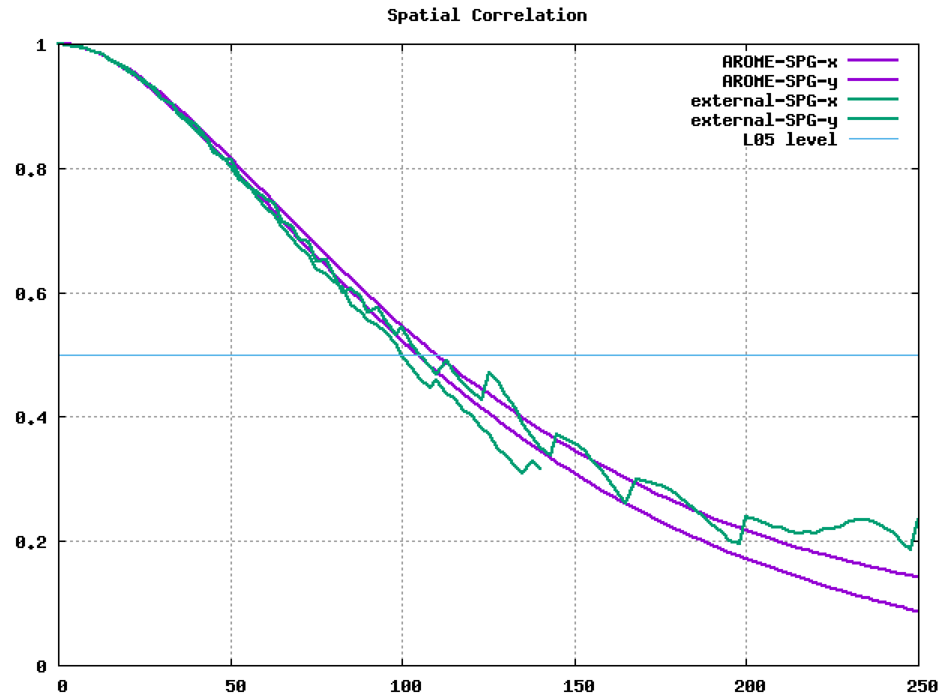

Fig. 6.: Spatial correlation functions calculated from fields of ALADIN code implemented SPG fields (purple) and from external SPG program results (green). Blue line shows the 0.5 level.

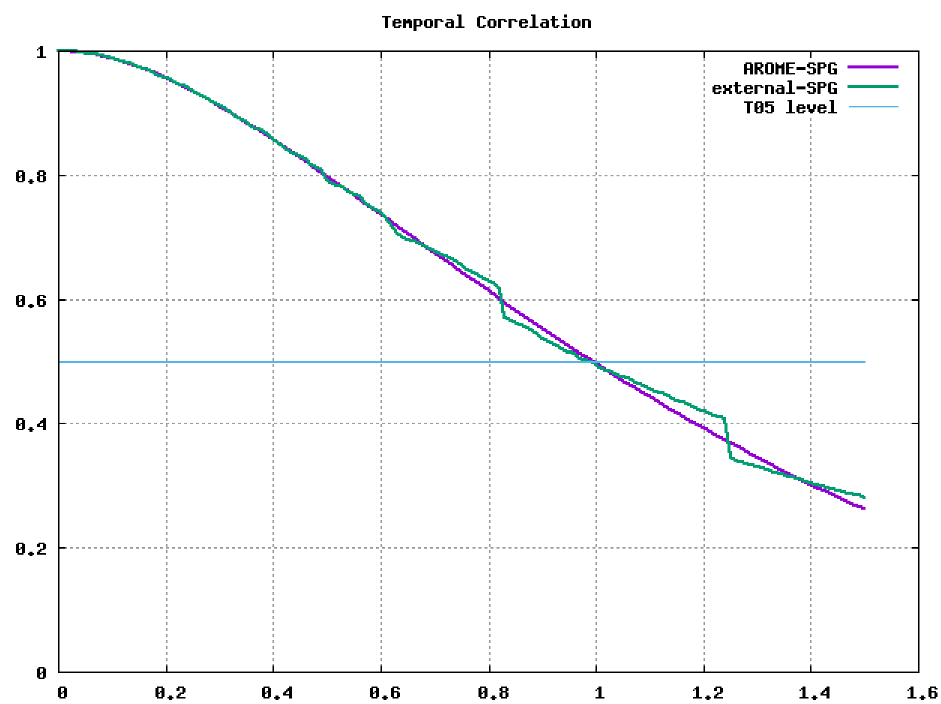

Fig. 7.: Temporal correlation functions calculated from fields of ALADIN code implemented SPG fields (purple) and from external SPG program results (green). Blue line shows the 0.5 level.

5. Conclusion and future plans

In section 1. it was underlined that current spectral pattern generator does not work properly in LAM. It means that its settings (standard deviation, horizontal correlation length) can not give back the expected results. This reason motivated that in section 2. the implementation of the Spectral Pattern Generator (SPG) was proposed. Some of its theoretically (eg. “Proporionality of scales”) and practically attractive property were highlighted. In section 3. some of the technical difficulties were described which I had to face with during my stay. In section 4. random pattern fields were visualized and its statistical behave was investigated. As a conclusion we can highlight that the ALADIN implemented version of SPG is able to give very similar results than the external program version. SPG works better than the current pattern generator from many aspects which can be highlighted as the following:
- SPG is better tunable (control parameters setting I more correct);
- SPG has the “proportionality of scales” property;
- SPG is closer to be really isotropic.

In my opinion SPG gives a good opportunity to use it in SPPT tests (no matter if we use partial or total tendencies, or if we tune the supersaturation check part). Even if there will be better schemes in the future than SPPT it is very likely that for model uncertainty representation NWPs will need random patterns which makes future for SPG in an after-SPPT world, as well. SPG can be useful also from surface perturbation aspects.
The current SPG implementation can have 3 direction to tune:
- Using L05 and T05 namelist parameters directly from the namelist. This would be just a very technical improvement which does not effect the performance of SPG but makes life of users much easier.
- Implementation of 3D version. It would make possible to give a vertical structure for random patterns which effect would be interesting to see.
- Implementation of non-Gaussian noise. It could be useful for perturb parameters which basically does not have Gaussian distribution (anything except temperature).

Acknowledgement

Finally I would like to say thank you for the whole modeling group of ZAMG, for their supportive attitude. They helped me in all my scientific, technical and personal troubles which made my work really smooth during my stay. I am also grateful to Martin Belluš for the inspiring discussions and mailings and for the patience what he showed as LACE Area Leader.
I am appropriate to Francois Bouttier, who supervises ECMWF's special project, called 'spfrbout', which SBU was used by me during my stay.
I am really thankful to Michael Tsyrulnikov and Dmitry Gayfulin from the Russian HydroMetCenter. I was allowed to examine their source program and documentations. Additionally they were very helpful and cooperative during our mailing.

References

Belluš, M., 2014: Stochastically perturbed physics tendencies of surface fields in ALADIN-LAEF system, Report on stay at ZAMG 12/05 - 20/06/2014, Vienna, Austria

Bouttier, F., Vié, B., Nuissier, O., Raynaud, L., 2012: Impact of Stochastic Physics in a Convection-Permitting Ensemble. Mon. Wea. Rev., **140**, 3706–3721.

Buizza, R., Miller, M., Palmer, T.N., 1999, Stochastic representation of model uncertainties in the ECMWF Ensemble Prediction System, Quart. J. Roy. Meteorol. Soc., **125**, 2887–2908.

Palmer, T., Buizza, R., Doblas-Reyes, F., Jung, T., Leutbecher, M., Shutts, G., Steinheimer, M., Weisheimer, A., 2009: Stochastic parametrization and model uncertainty. Tech. Rep., ECMWF Tech. Memo. **598**, 42 pp. [Available online at http://www.ecmwf.int/publications/.]

Szűcs, M., 2014: Stochastically Perturbed Parameterized Tendencies in ALARO and AROME , Report on stay at ZAMG 06/10/2014 - 31/10/2014, Vienna, Austria

Szűcs, M., 2015: Tests of possible SPPT developments , Report on stay at ZAMG 28/09/2015 - 06/11/2015, Vienna, Austria

Szűcs, M., 2016a: Stochastic pattern generators , Report on stay at ZAMG 23/05/2016 - 17/06/2016

Szűcs, M., 2016b: Test of the SPPT scheme, Joint 26th ALADIN Workshop & HIRLAM All Staff Meeting 2016, 4-8/04/2016, Lisbon, Portugal

Szűcs, M., 2016c: The convection-permitting ensemble system of the Hungarian Meteorological Service, SRNWP-EPS II Workshop “Probabilistic prediction of severe weather phenomena” 17-19 May 2016, Bologna, Italy

Szűcs, M., 2016d: SPPT in AROME and ALARO: test reuslts and open questions, HIRLAM EPS Working Week 2016-2
https://hirlam.org/trac/attachment/wiki/Meetings/EPS/EPS2016-2/SPPT_SzucsMihaly.pdf

Tsyrulnikov, M. and Gayfulin, D., 2016: A Limited-Area Spatio-Temporal Stochastic Pattern Generator for ensemble prediction and ensemble data assimilation, Meteorologische Zeitschrift, .