Stochastic Pattern Generator (standalone version)
User's guide

D.Gayfulin, M.Tsyrulnikov

HydroMetCenter of Russia

(gayfulin@rambler.ru, mik.tsyrulnikov@gmail.com)

11 November 2016

1 Structure of the package

1.1 Overview

The stochastic pattern generator (SPG) package is located in the folder `SPG/`. This folder contains three subfolders: `Config/`, `Obj/`, `Src/` and some files in the main folder. In this section we give a brief description of all folders and files of the package.

The `Config/` folder contains two configuration files: `testrun.cfg` and `operational.cfg`. These files contain setup parameters of the generator. The configuration file `testrun.cfg` is created for the test run of the generator. Please do not change it. In the configuration file `operational.cfg` you can change the setup parameters as you wish under the constraints described in section 3.

The `Doc/` folder contains the documentation of this package.

The `Obj/` folder contains `*.o` object files used for linking of the main executable file `SPG/main.exe`.

The folder `Src/` contains files with the source code of the package. The main program file is called `main.f90`. The program also uses two module files. The module file `set_gen_params.f90` contains subroutines that are used to calculate the internal parameters of the generator. In the module file `set_gen_params.f90` we compute the pseudo-random field. In the SPG package we also use some external freely distributable modules. The module file `fft99f.f` performs the backward fast Fourier transform. The folder `rng/` contains IMSL subroutines that are used to generate the white noise.

1.2 Files located in the main folder

There are two compilation scripts: `icompile` and `gcompile`. The first script uses the Intel ifort compiler, while the second script uses gfortran. Each scripts creates the main executable file `main.exe`.

The file `main.exe` is launched using the script `launch.sh` or `test_launch.sh`. The former script uses the configuration file `operational.cfg` and writes the output diagnostic information to the file `out`. The latter script uses another configuration file `test_run.cfg` and writes the output diagnostic information to the file `test_out`.

The etalon version of testrun output is called `testrun_etalon_out`. Before using this package, please launch the SPG using the script `test_launch.sh` and compare the file `test_out` with `testrun_etalon_out`. These files should coincide.

2 Outline of the computational algorithm

The program starts with reading the configuration file. Then it calculates the internal parameters of the generator and allocate the fields. After that it computes the pseudo-random field. This process has two main stages: integration in spectral space and fast Fourier transform that computes the field in physical space. When the computation of the random field is completed, estimates of its spatial (in horizontal and vertical dimension) and temporal covariance functions, are to computed.

Note that the integration is performed on 3D (or 2D in the two-dimensional version) torus. In order to avoid correlations between the opposite sides of the spatial domain, we perform integration on the grid with greater number of points. For instance, instead of the domain with $256 \times 256 \times 32$ points, we generate the random field on the domain with $300 \times 300 \times 36$ points and then take the desired field from the extended domain.

3 SPG setup

Setup parameters of the SPG are set up in the configuration file. The configuration file name for the current program run is specified in the script `launch.sh`. You can modify the setup parameters in the configuration file under the following restrictions:

1. **ngrid** – the gridsize in each of the horizontal coordinates, should be a positive integer number.

2. **vertngrid** – the vertical gridsize. If you want to generate the 2-dimensional field, **vertngrid** should be 0 or 1, otherwise should be a positive integer number greater than 1.

3. **timestride** – the time stride of the SPG. i.e. the time interval between the subsequent output spatial fields.

4. **nstride** – the number of time strides. Should be a positive integer.

5. **nsample** – number of independent samples of the field to be generated. Should be a positive integer. Different samples are uncorrelated.

6. **std** – the random field's standard deviation. Should be a positive real number.

7. **L05** – the spatial length scale at the correlation level 0.5. Should be greater than **mesh_size** (see below) and less than **0.5*ngrid*mesh_size**.

8. **T02** – the temporal length scale at the correlation level 0.2. Should be greater than **timestride**.

9. **a_dt_min, a_dt_max** – spectral time step parameters. The greater **a_dt_min** and **a_dt_max** are, the faster (though less precise) the SPG is. We recommend to use the default values 0.05 for **a_dt_min** and 3 for **a_dt_max**. **a_dt_min** should be less than 0.25 and **a_dt_max** should not exceed 5.

10. $\mathbf{n_0}$ – n_0 parameter of the coarse grid in spectral space. The greater n_0 is, the more precise (though slower) the SPG is. n_0 should be greater than 20 and should not exceed **ngrid**/2.

11. $\boldsymbol{\varepsilon}$ – ε parameter of the coarse grid in spectral space. The greater ε is, the less precise (though faster) the SPG is. ε should be greater than 0.01 and should not exceed 0.2.

12. **mesh_size** – horizontal grid spacing (km). Should be a positive real number.

13. **dseed** – random number generation parameter. Should be a real number, from 1 to 2147483647 .

4 Random field

The generated random field is available from inside `main.f90` under the name `rand_field`. It has five indexes:

1. The horizontal coordinate x. The extents are $(1 : ngrid)$.
2. The horizontal coordinate y. The extents are $(1 : ngrid)$.
3. The vertical coordinate z. If the SPG is launched in the 2-dimensional mode, this coordinate is not used, the extents are $(0 : 0)$, otherwise the extents are $(1 : vertngird)$.
4. The temporal coordinate t. The extents are $(1 : nstride)$.
5. The number of samples. The program can simultaneously generate several independent fields. The extents are $(1 : nsample)$.

5 Output diagnostics file

The output diagnostics file of the SPG is called `out`. It has the following structure: first of all, the setup parameters from the configuration are listed. Then, parameters of the extended grid are printed. After that, you can see the internal parameters of the SPG and then the information about the integration in spectral space and fast Fourier transform. Finally, you can see the statistics of the generated field: standard deviation, minimal and maximal values, and also the spatial and temporal correlation functions.

6 Conclusions

For the detailed mathematical description of the SPG see arxiv.org/abs/1605.02018.