SkyGlow - installation manual

The present version of SkyGlow is only for tests purposes and for those who want to become familiar with this tool. SkyGlow is not dedicated for commercial use, so any potential user should accept that the product features are partly limited. Our capacities for programming are restricted, but we appreciate any suggestions for improvements of the present version of SkyGlow. We strongly encourage those users who are interested in this product to send us comments that can help to improve next versions of SkyGlow.

This version contains much faster routine for clear-sky conditions calculations. The light sources surrounding the hypothetical observer can be searched and loaded automatically from the database.

Part of SkyGlow is written in Java, so it is platform independent and no special installation is needed. You have simply to unpack the SkyGlow.zip file to any folder on your computer and run SkyGlow.jar program. But, SkyGlow uses other tools (e.g. GNUplot visualizer), so some explanation is needed for various operational systems. Following chapters explain how to:

- Install and run the SkyGlow software on Microsoft Windows
- Install and run the SkyGlow software on Linux
- Make basic tests of the SkyGlow software

The computational core is compiled for Intel-compatible hardware platform. So, it will not run on other platforms, like ARM-based tablets etc. We are working on a platform-independent computation core (in Java), which will be available in one of next releases. This release contains also a core compiled for OS X, but it is only for testing purposes.

Installing and running the SkyGlow software in Microsoft Windows

Program was successfully tested under Windows 7 (both 32 bit and 64 bit versions), Windows platforms 8 and 8.1. and Windows 10.

Please, follow next steps:

1. in case you don't have GNUplot already installed: you can download and install it from the web-site
   http://sourceforge.net/projects/gnuplot/files/gnuplot/
   We have made few tests with the current version 5.0.4 (**gp504-win32-setup.exe**).

2. in case you don't have Java Runtime Environment already installed: you can download and install it from the web-site
   http://java.com/en/download/

3. if the only thing you need to do is to unpack the **SkyGlow.zip** file: You can unzip the file to any folder on your computer with permissions to create the files (everybody should have the read/write rights in "My Documents" or "Desktop" folders). Afterwards you should run (double-click) the file
   SkyGlow.jar

Installing and running the SkyGlow software in Linux

Program was successfully tested in current version of Kubuntu (16.04 LTS) and should run correctly for other distributions too.

Please, follow next steps:

1. in case you don't have GNUplot already installed: type following command in terminal (Konsole, Terminal or similar):
   sudo apt-get install gnuplot-x11

2. in case you don't have Java Runtime Environment already installed: type following command in terminal (Konsole, Terminal or similar):
   sudo apt-get install default-jre

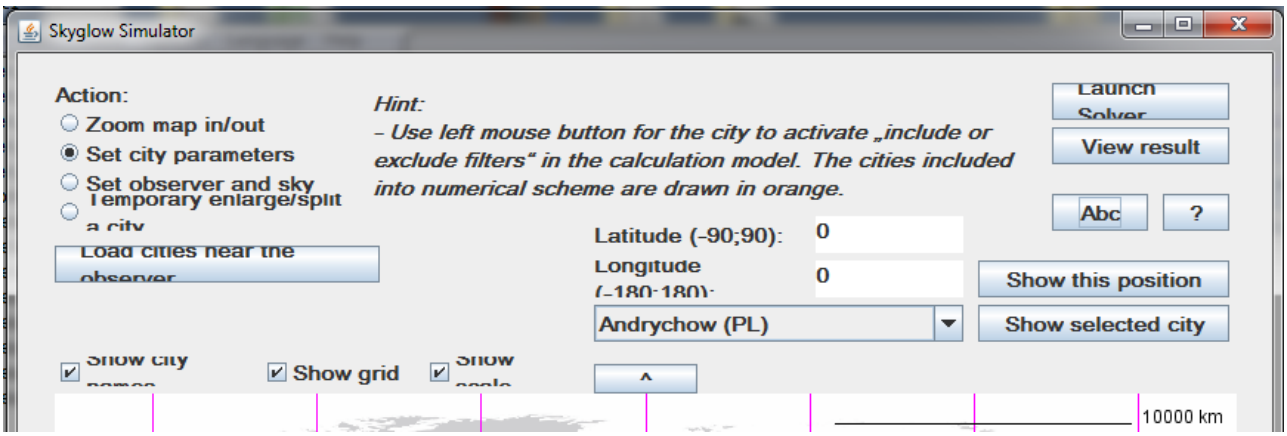
3. in case you don't have **xterm** terminal emulator already installed: type following command in terminal (Konsole, Terminal or similar):
   sudo apt-get install xterm

4. unpack the **SkyGlow.zip** file to any folder on your computer, where you have permissions to create own files (e.g. somewhere in "home" folder).

5. change the **Skyglow.sh**-file permissions – mark this file as executable. Afterwards run **SkyGlow.sh** by clicking it by mouse.
   Another possibility is to type following command on terminal (Konsole, Terminal or similar):
   sh SkyGlow.sh

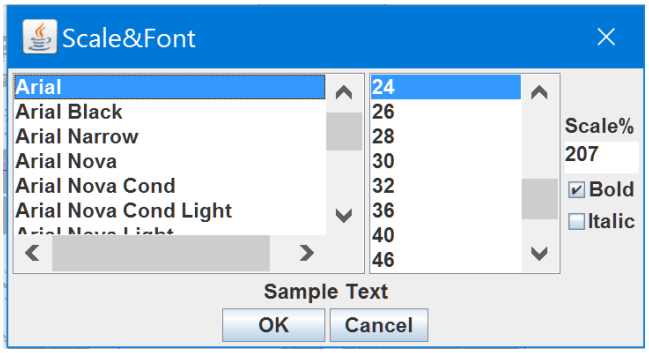
Basic tests of the SkyGlow software

Although Java is platform-independent, a use of fonts and screen resolutions is manifold on different platforms. Various computers can have various fonts installed and the look of java-programs can change.
The font "Arial" is present by default in all versions of Microsoft Windows, so we designed our software for this font and the look of the SkyGlow software should be the same in all versions of Microsoft Windows. In Kubuntu we have used the font "NanumGothic" (installed by default) with similar face.
But we cannot guarantee that these fonts are present in your computer. If the program looks like in following figure, you have to select another font from list of available fonts and/or resize the program window.

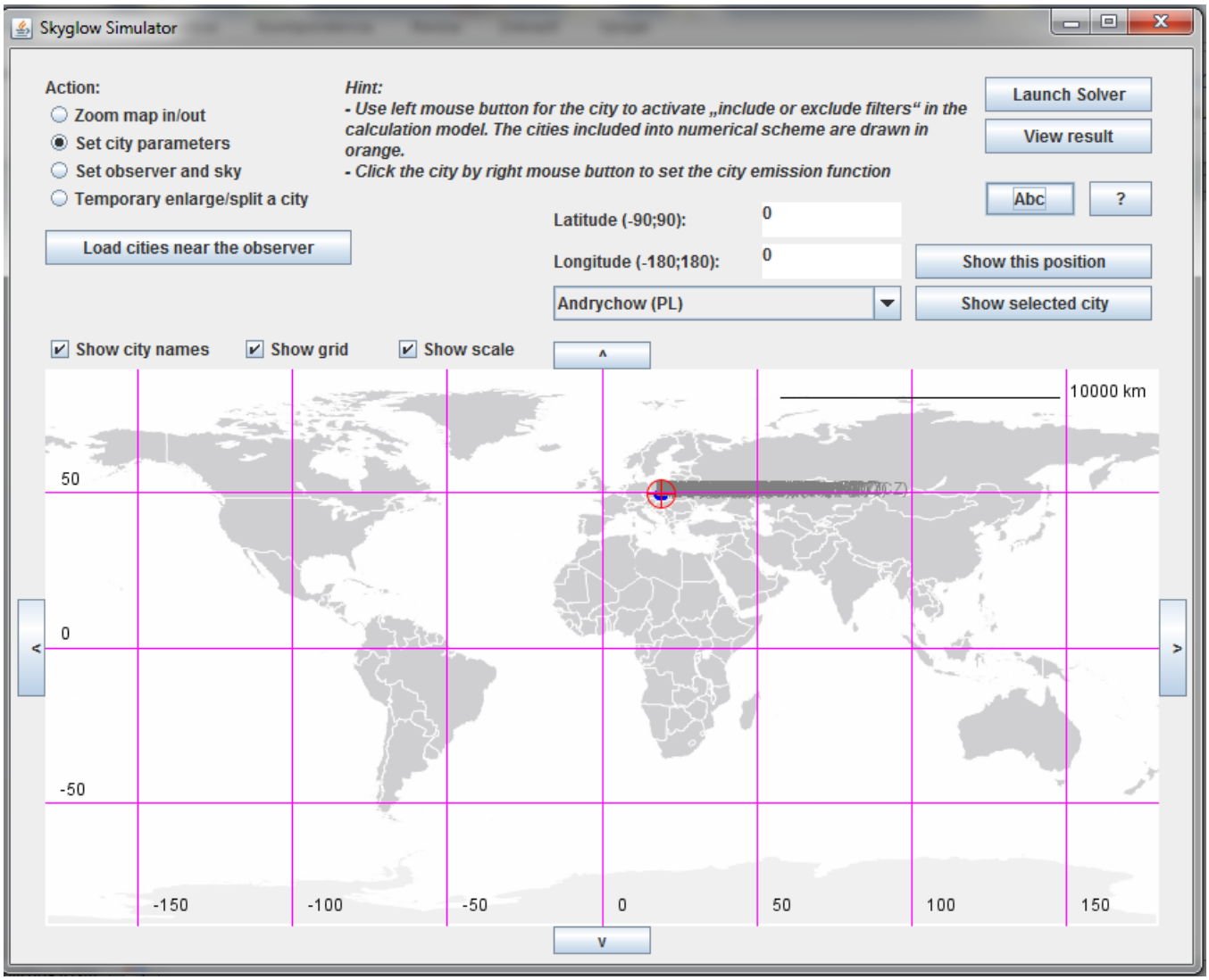


Improper font: texts are too wide and do not fit the reserved space (e.g. in buttons)

In first step, resize the SkyGlow-window to an appropriate size. Then click the "Abc" button and find a suitable font. You can also experiment with the size of the font (usually 11 or 12 is the best). Try also "bold" modification. Several font families include also "Condensed" modification of the font. Selected font is saved and will be used in future runs of the program. If needed, you can choose/save another font in any time. You can also adjust the Scale-factor – the size of the program window:



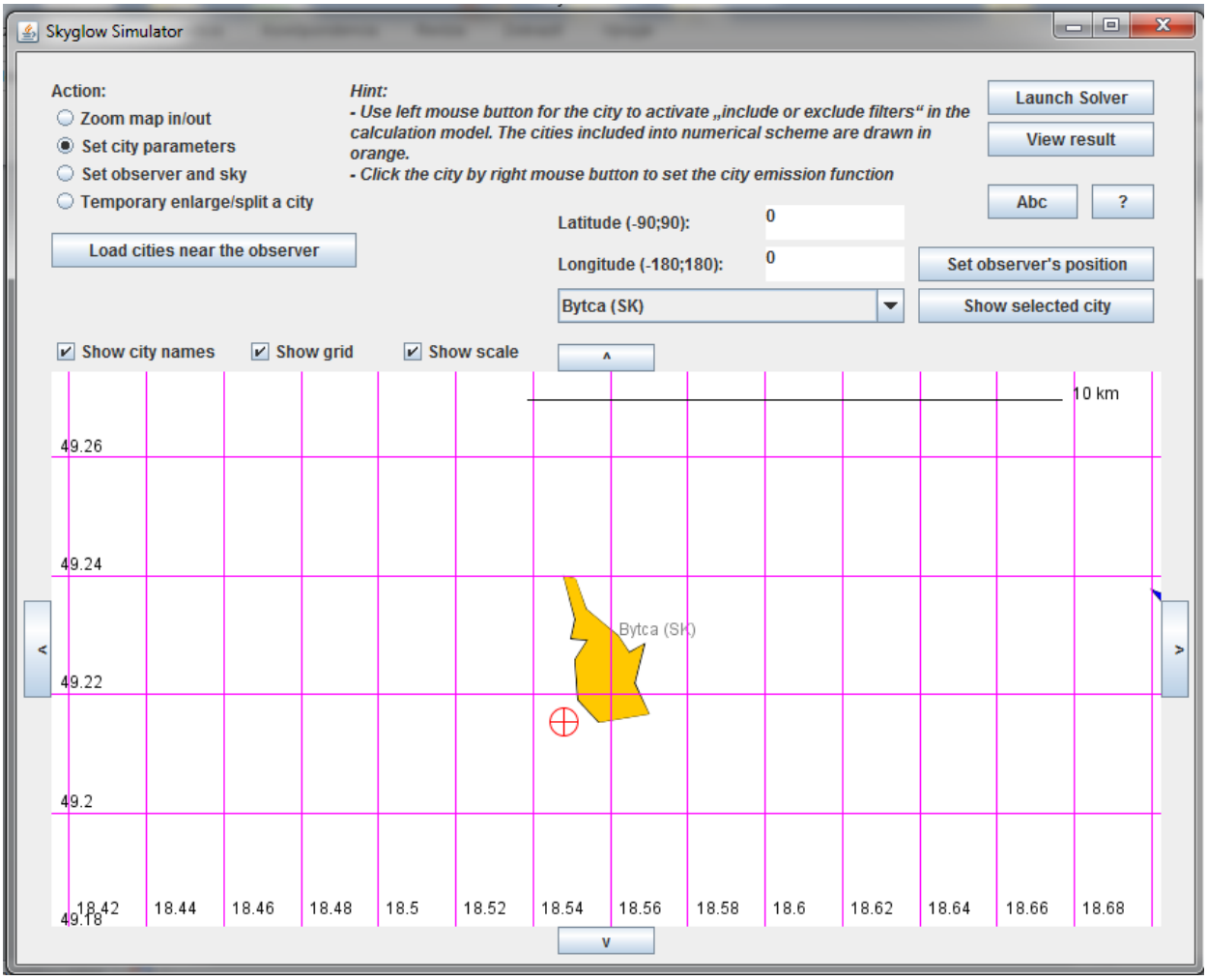
Proper look of the program is as follows:



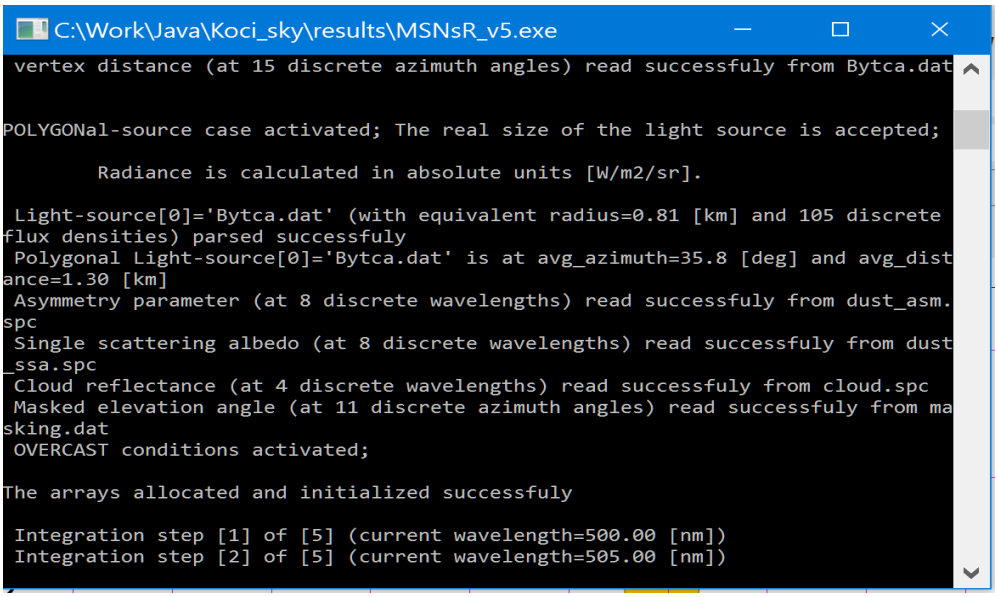
Proper look of the program

The default model is to be used when no input parameter will be modified.

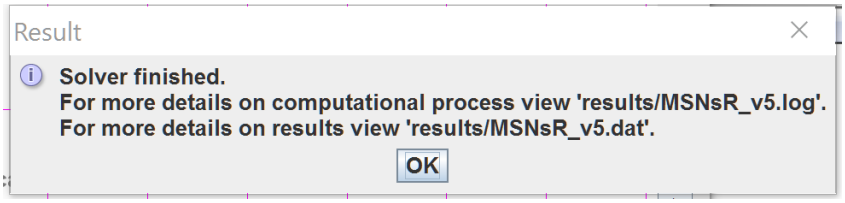
Click "Show selected city" button, then click a city closest to the observer (red cross) by left mouse button to switch it on (the city will get orange color). You will see picture like below (don't worry if another city is pre-selected):



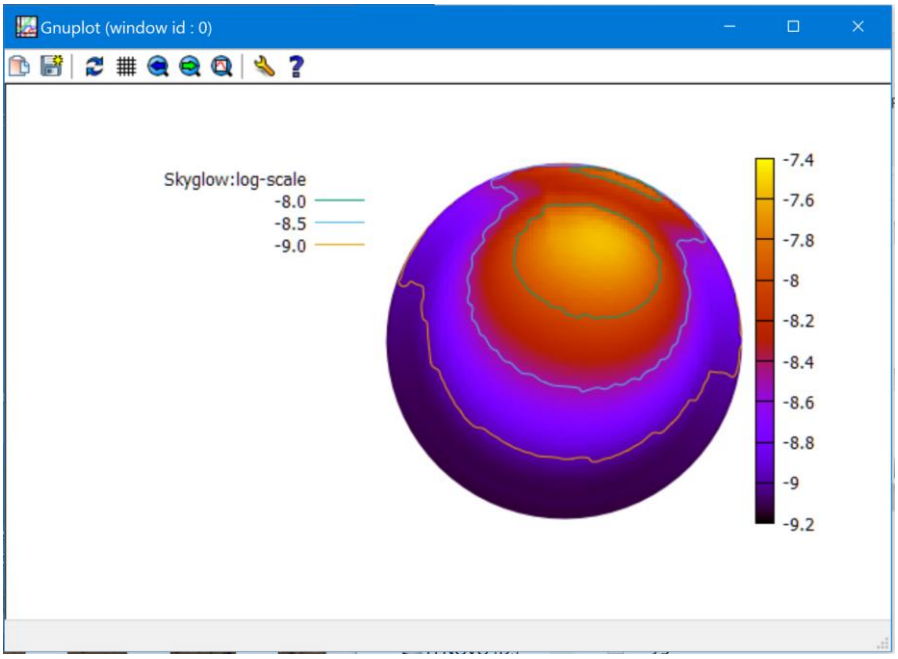
Now try to start the computations. Click the "Launch Solver" button. The solver (singleton written in C++) will start in a console, e.g. in Windows:



The following message is displayed after the solver has finished its job:



Select OK and click the "View result" button. The GNUplot will be instantiated and you will see the figure like below (the picture depends on the city pre-selected):



Computational details are printed to the file "results/MSNsR_v5.dat". In the "results" folder you can find also Portable Network Graphics (PNG) and Enhanced Metafile (EMF) versions of the picture shown by GNUplot.

The software allows you to make following changes in the computational model:

- switch any city on/off
- change the parameters of the city emission function (e.g. spectrum or radiant intensity as a function of zenith angle) and other parameters of the city selected
- use a clear sky model or various overcast conditions
- change the position of an observer
- search and reload light sources surrounding a hypothetical observer (the data are read from the folder entitled "cities" and from all underlying subfolders)
- change location-specific parameters (aerosol content and type, …)
- temporarily split or enlarge the city to simulate the city reconstruction
- change the spectral band or switch between the output quantities: photometric or radiometric

To set the parameters of the model, select the corresponding action using radio-button and then use the mouse (see "Hint" area of the screen).

The present version of the program uses only few (test) models for aerosol particles, clouds, cities, … We strongly encourage the users to create their own models and share it with whole community and send copies of

new files to m.kocifaj@gmail.com. This is the best way how we can build the database for many regions worldwide. The file-structure we have built till now is self-explanatory. Some meta-data can be found in the header part of each file (# <keyword> free text). Two-column data in *cty files are used for geographical latitude (1st col) and longitude (2nd col) of all vertices that define the polygonal map of a city. At present, only two city emission functions (cefs) can be used (cosine & Garstang's). Data input is currently under development. Spectra of light sources can be defined in cefs*.spc files (the number of data is limited to 200). Spectral profiles of aerosol asymmetry parameter (*asy) and single scattering albedo (*.ssa) can be find in directory "aerosols". The number of data is limited to 200. The same is true for spectral albedo of cloudy layer (see clouds*cld). The obstacles are simulated through horizon shielding – see the files horizon*.msk where the first column is the azimuth while the second column is the elevation angle of an obstacle.

Note 1: All possible options are loaded during the start of the SkyGlow program. If you change the content of any configuration file (e.g. albedo of an aerosol) or create a new file (e.g. new spectral dependence of CEF), you have to restart the program.

Note 2: Setting the clear-sky conditions (many layers of the atmosphere have to be considered) and/or the photometric output (many wavelengths have to be calculated) can lead to time-expensive calculations. This will change later when multithreading application will be released.

Note 3: At present, only two city emission functions (CEFs) can be used (Cosine and Garstang). Data-input is under development.

Troubleshooting the problems:

There are two log-files generated by the computational modules: "logfile.txt" and "results/MSNsR_v5.log".

The first one stores the SkyGlow outputs, such as information on your settings and actions performed before solver (or viewer) was launched.

The second one contains logs from core computations.

Both log-files can be useful in searching the source of eventual problems. The core-logs are now heavily shortened to save the disk space. The full log is only activated in coding phase and should help developers in bug-fixing.