SkyGlow 6.1 - 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 parallelized computational core. It uses specialized MS Windows libraries. The input configuration- and data-files are in more flexible XML format that allows for easy future extending.

GUI of SkyGlow is written in Java (it is platform independent), so you have to install the last version of the Java Runtime Environment or the last version of the Java Development Kit. Then you have simply to unpack the SkyGlow.zip file to any folder on your computer and run SkyGlow.jar program. However, SkyGlow uses GNUplot visualizer for the graphical presentation of results, and thus also GNUplot must be installed.

Contents

1. Installing and running the SkyGlow 6.1 software in Microsoft Windows 2

2. Basic settings of the SkyGlow software 2

3. Basic test of the SkyGlow software 4

4. Running the computation without GUI 6

5. Changing parameters of the model 7

   Including cities into calculation 7

   Changing the city emission function 8

   Setting the position of the observer 9

   Changing the local parameters (atmosphere, horizon) and specifying the outputs 10

   Displaying the results 12

   Splitting/enlarging the city 13

   Setting the precision/resolution 15

6. Creating own model 16

   Creating the city-files 16

   Creating the spectra of lights 18

   Mixing the spectra of lights 19

   Creating the spectral reflectivity of clouds 20

   Creating the horizon profiles 20

   Creating the aerosol SSA and ASY files 20

7. Troubleshooting the problems 20

1. Installing and running the SkyGlow 6.1 software in Microsoft Windows

Program was successfully tested under Windows 10/11 (both 32-bit and 64-bit versions). Please, follow next steps:

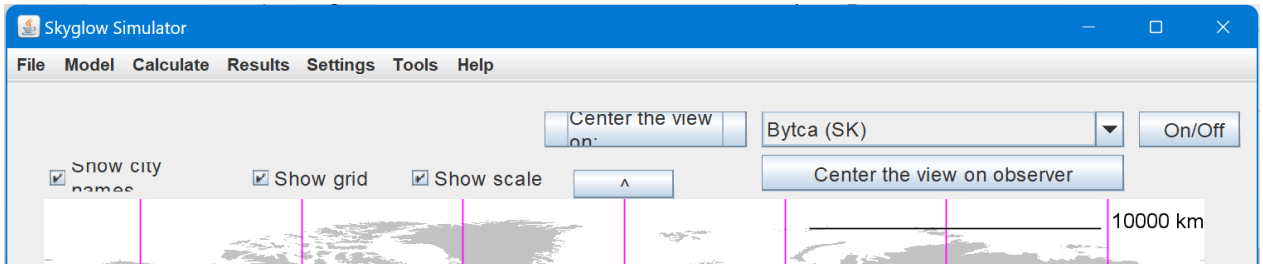
1. In the case case you don't have GNUplot already installed: you can download and install it from the web-site
   http://sourceforge.net/projects/gnuplot/files/gnuplot/
   We have made successful tests with the current version 5.4.8 (**gp548-win64-mingw.exe**). Please, do not disable the association of *.PLT files with GNUplot during the installation. After correct installation, the GNUplot is started if any PLT-file is double-clicked.

2. In the 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/
   We have made successful test with the current Version 8 Update 391. After correct installation, Java is started if any JAR-file is double-clicked.

3. Unpack the **SkyGlow.zip** file: You can unzip the file to any folder in your computer where you have permissions to create the files (e.g. "My Documents" or "Desktop" folder or any subfolder). Afterwards you should run (double-click) the file
   SkyGlow.jar

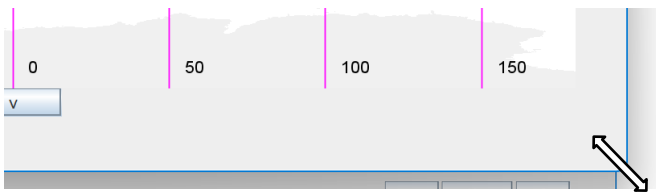
2. Basic settings of the SkyGlow software

Although Java is platform-independent, the 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.
But we cannot guarantee that these fonts are present in your computer. If the program looks like in following figure (texts are too wide and do not fit into reserved space, e.g. in buttons), you have to select another font from list of available fonts.



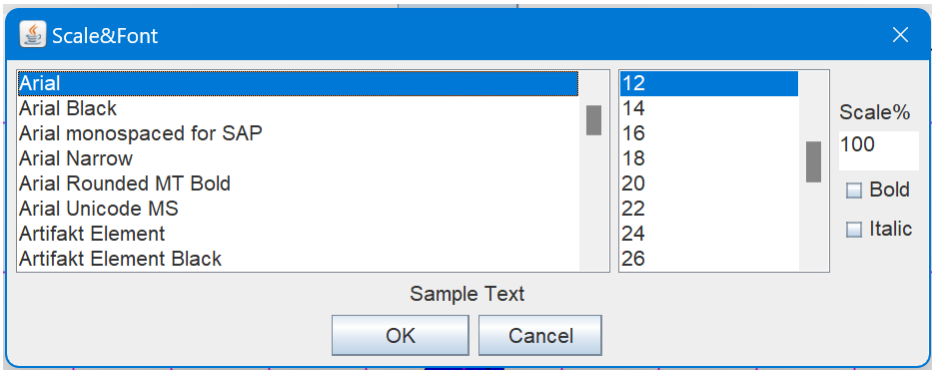
Improper font: texts are too wide and do not fit the reserved space (e.g. in buttons)

In first step, resize the SkyGlow-window to an appropriate size by dragging the right-bottom corner of the window:



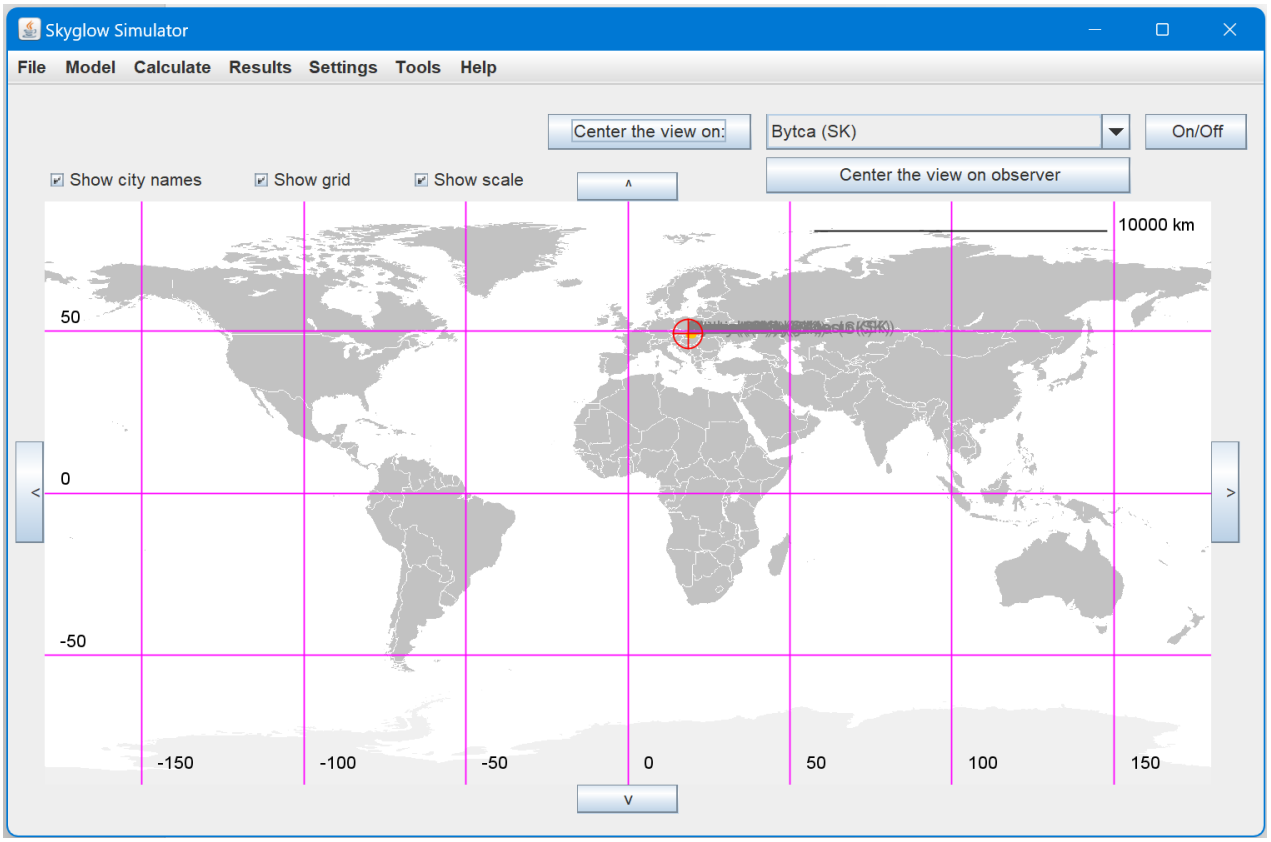
Changing the size of the application window

Then use the “Settings/Font & Application size” menu item 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” or “italic” modifications. Several font families include also “Condensed” or “Narrow” 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. The “Scale%”-parameter controls the size of the application window.

Proper look of the program is as follows:

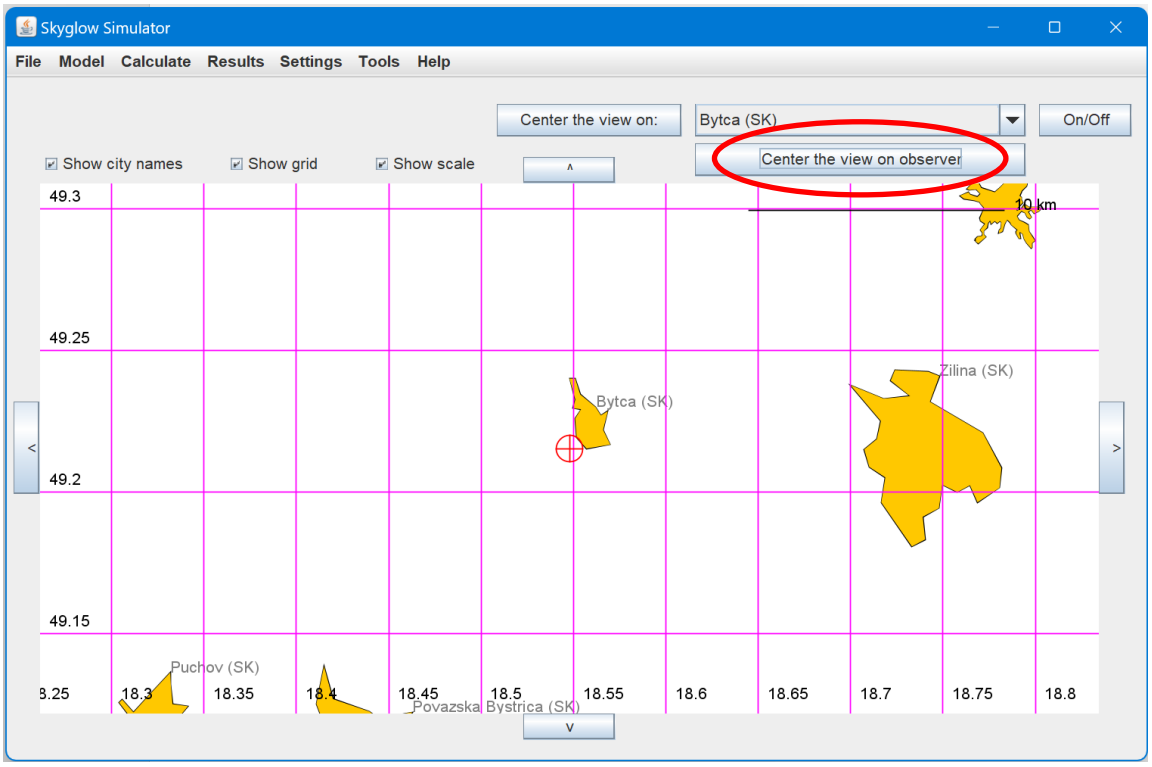


Proper look of the program

In the case you are not happy with the pixel resolution of the application (the content of the application window is sometimes not sharp on high-DPI screens), right-click the “SkyGlow.exe” launcher, go to submenu “Properties/Compatibility” and click the “Change high DPI settings”. In the dialog check the “Override high DPI scaling behavior. Scaling performed by: Application”. Click twice OK to apply the changes. Use the “SkyGlow.exe” launcher to start the SkyGlow software and resize the SkyGlow-window to appropriate size.

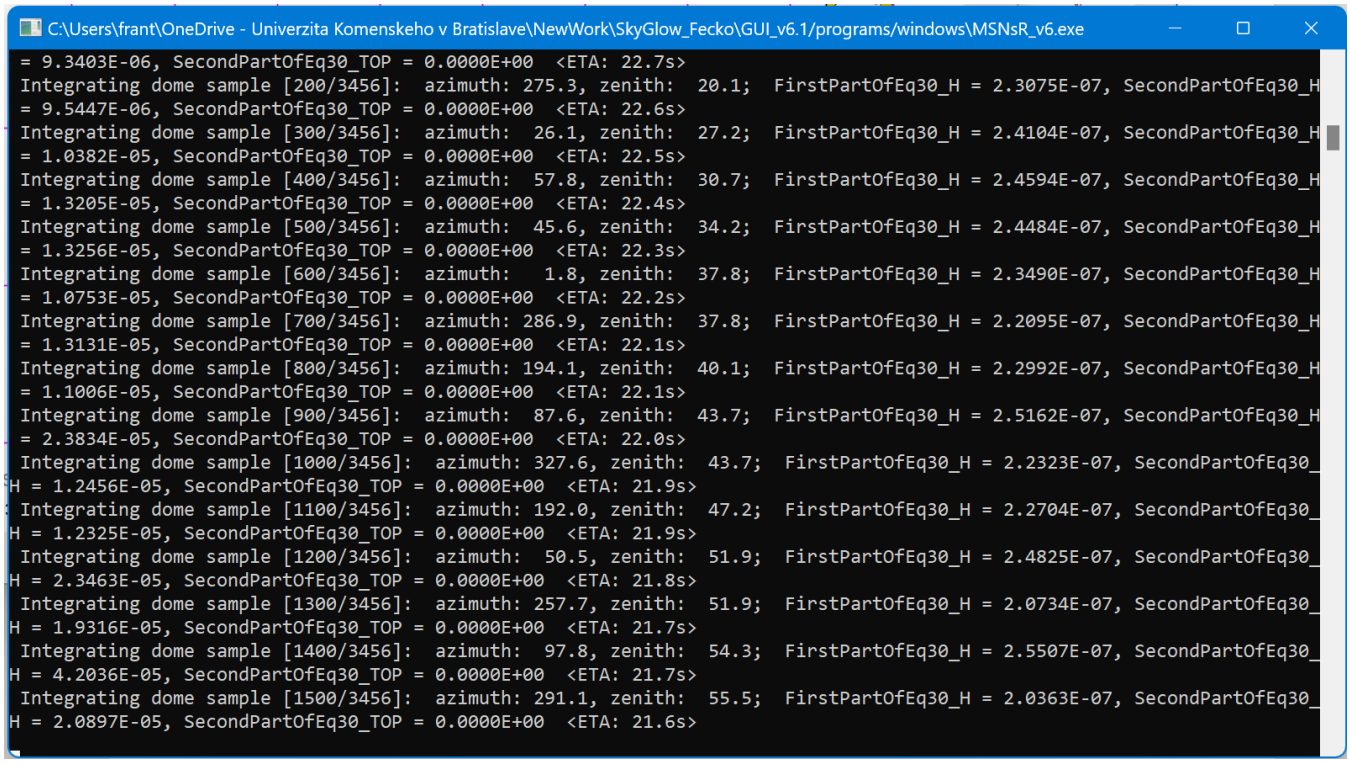
3. Basic test of the SkyGlow software

Click "**Center the view on observer**" button to see the cities close to the observer (red cross). You will see picture like below:



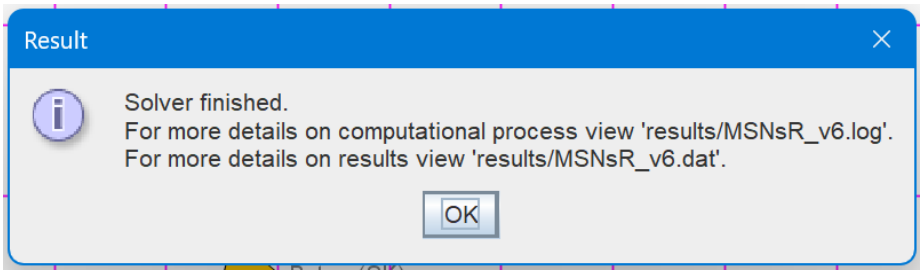
The default parameters of the model (city, sky) are used, if not changed by the user. So, we can start the calculations immediately.

Select the "**Calculate/Save project & Solve**" menu item. The solver (singleton written in C++) will start in a console:



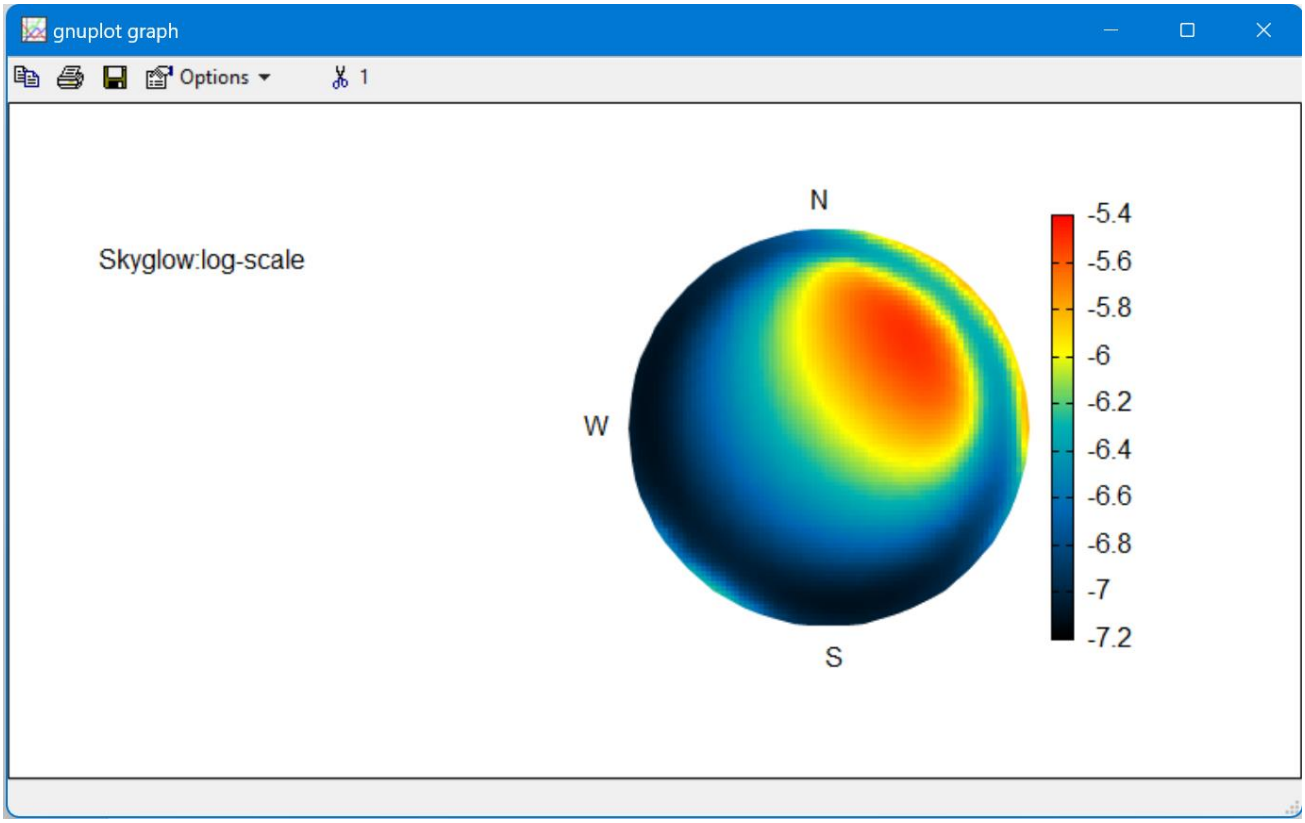
Please, note the ETA value (the estimated time to finish the job).

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



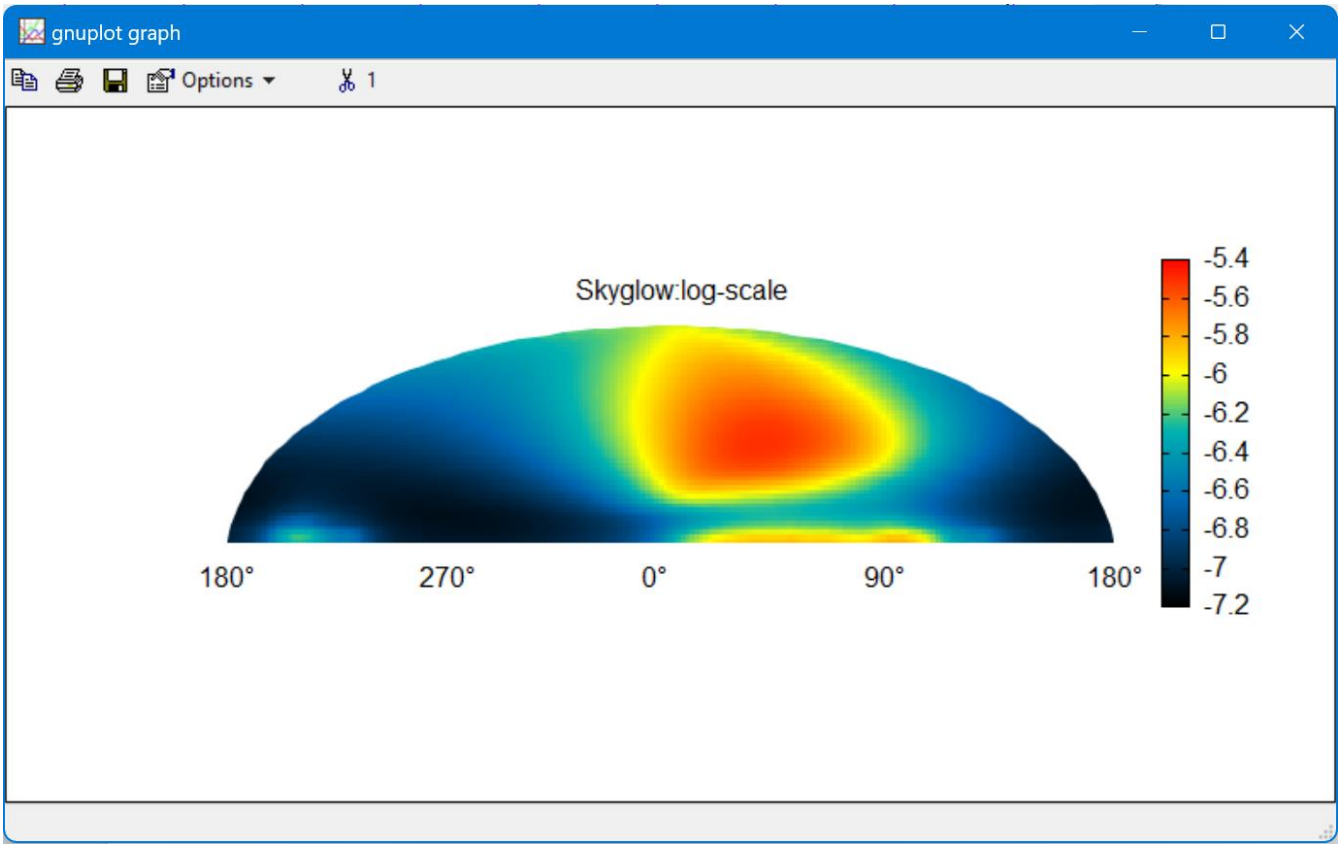
As you can read, more details about computations can be found in the file “MSNsR_v6.log” located in “results” folder. More results (computed values of the skyglow and main parameters used) are located in “MSNsR_v6.dat” file.
Select OK.

Then select the "**Results/Polar plot (God’s view)**" menu item. The GNUplot will be started and you will see two figure like below. This figure represents calculated skyglow (logarithmic value) in polar-plot (God’s view). Such a representation is very suitable for the comparison of the calculated pattern with the map of cities.



Skyglow in polar-plot (God’s view)

Then select the "**Results/Hammer projection (Observer’s view)**" menu item. This second figure represents calculated skyglow (logarithmic value) in Hammer-projection (Observer’s view). Such a projection shows more details in the parts of the sky close to the horizon and less details of the sky close to the zenith (where usually only small changes are observed).



Skyglow in Hammer-projection (Observer's view)

Portable Network Graphics (PNG) versions of these pictures shown by GNUplot can be found in the "results" folder.

Computational details are printed to the "results/MSNsR_v5.dat" file. This file can be shown using "Results/More results" menu item. The calculation logs are stored in "results/MSNsR_v5.log" file. This file can be shown using "Results/Log-file" menu item.
All parameters of the calculation are stored in "results/MSNsR_v6.cfg" file. You can load these parameters using "File/Load last saved project parameters" menu item.

4. Running the computation without GUI

The calculation (eventually after changing the parameters in the CFG file) can be re-run without using GUI by double-clicking the "results/run.cmd" file.
If the CFG-file was changed and the calculation was re-run without GUI, copy the tools "Results2gnuplotTrianglesPolar.jar" and "Results2gnuplotTrianglesHammer.jar" from "programs/windows" subdirectory to "results" subdirectory and start them. The files "results_polar.dat" and "results_hammer.dat" will be re-created from the "MSNsR_v6.dat" file.
The PNG file can be re-created by double clicking corresponding PLT or CMD file.

5. Changing parameters of the model

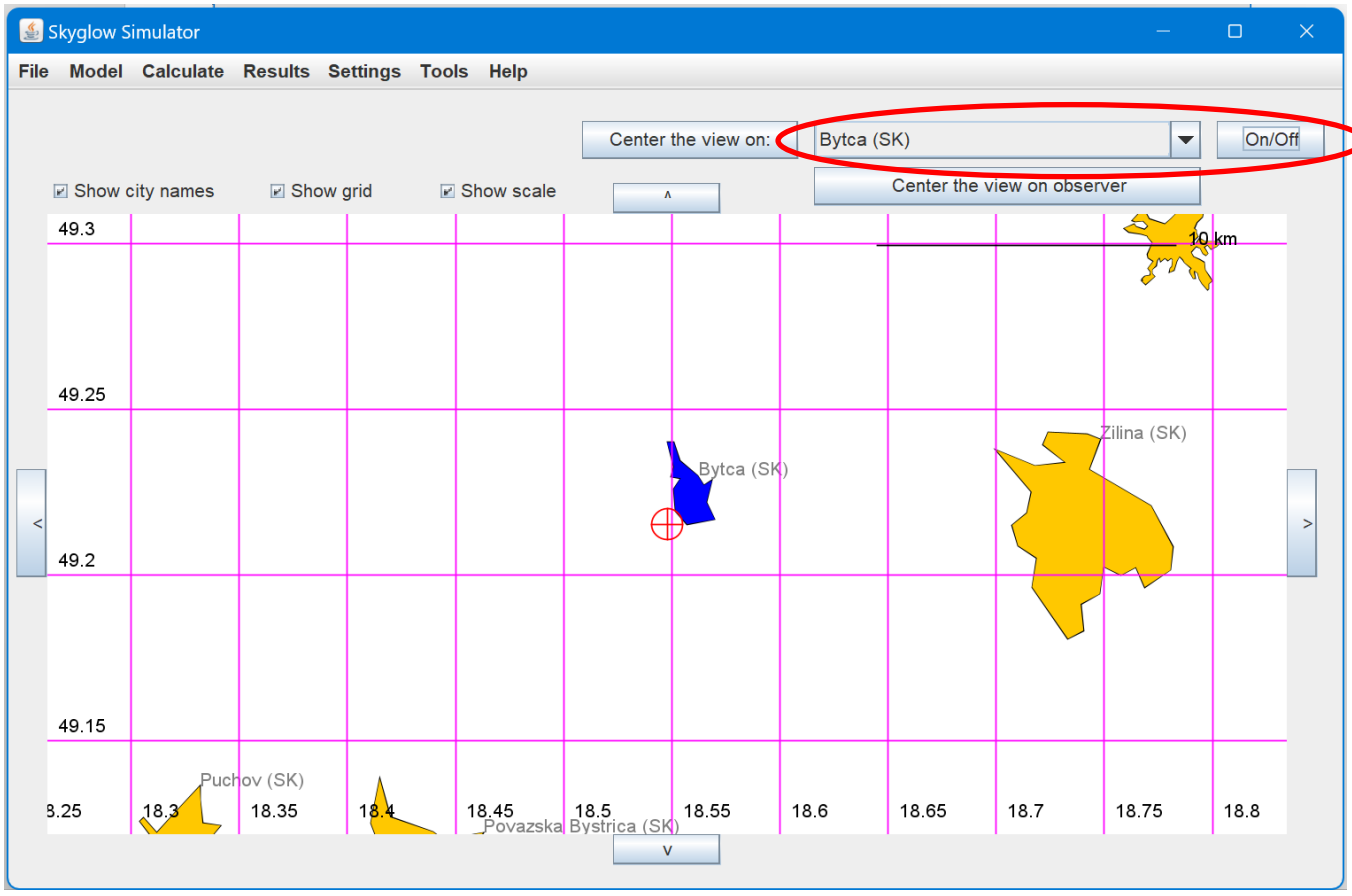
The GUI allows you to make following changes in the computational model:

- switch any city on/off (include/exclude it from the calculation)
- 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 (including partly cloudy condition)
- change the position of the observer and the direction of observer's view in Hammer projection
- change location-specific parameters (aerosol content and type, …)
- change the output spectral band or switch between the output quantities: photometric or radiometric
- split or enlarge the city to simulate the city reconstruction

To set the parameters of the model, select the corresponding "Model"-menu item or use the mouse (see "Help/Quick mouse actions" for details).

Including cities into calculation

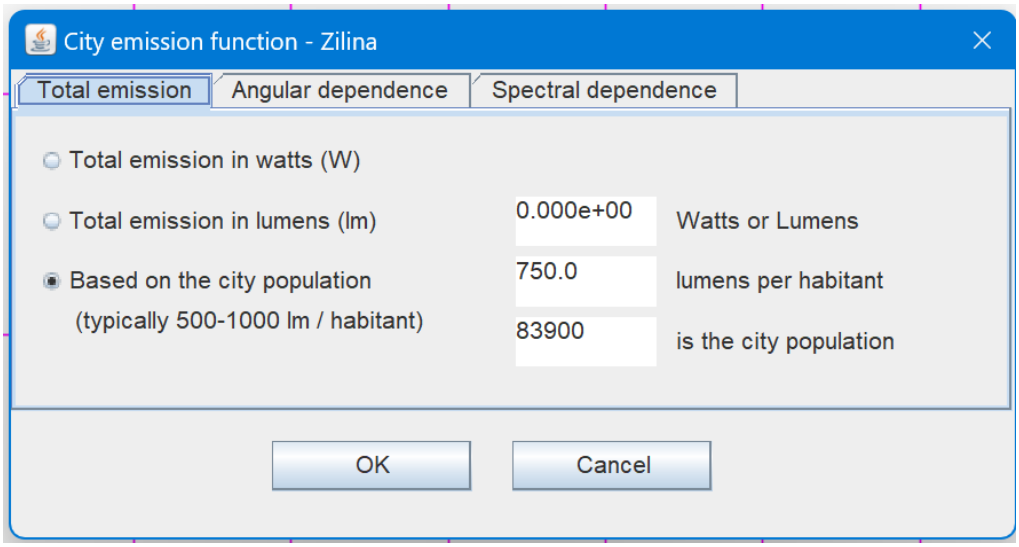
To include/exclude an individual city from the model, **left-click the city** (use mouse wheel to zoom in/out the map). The city will change its color from orange (included) to blue (excluded) and vice-versa.



Alternatively, select corresponding city in the combo-box and click the "On/Off" button. Menu items "Model/Include all cities" and "Model/Exclude all cities" allow to include/exclude all cities at once.

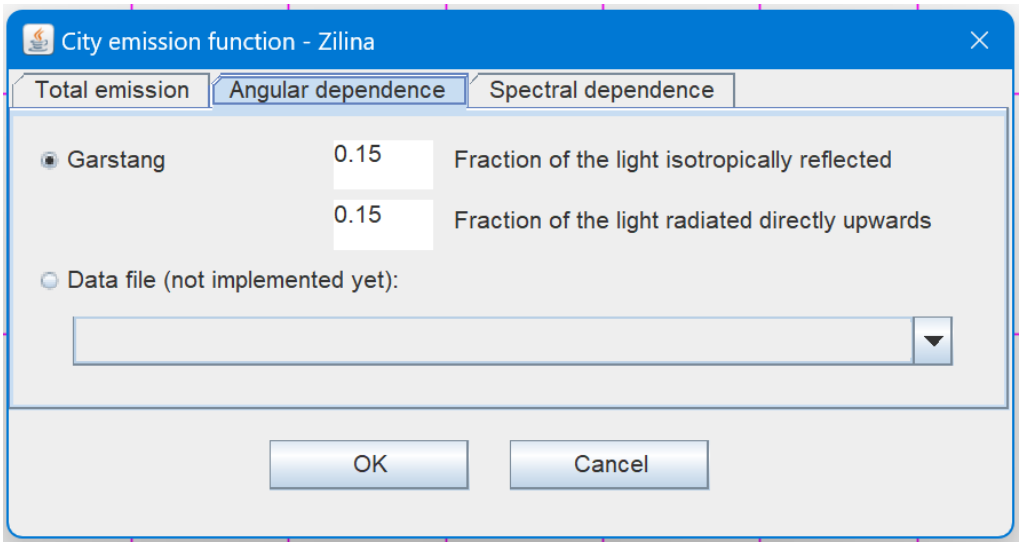
Changing the city emission function

To change the parameters of any city, **right-click** it. A dialog will be displayed:

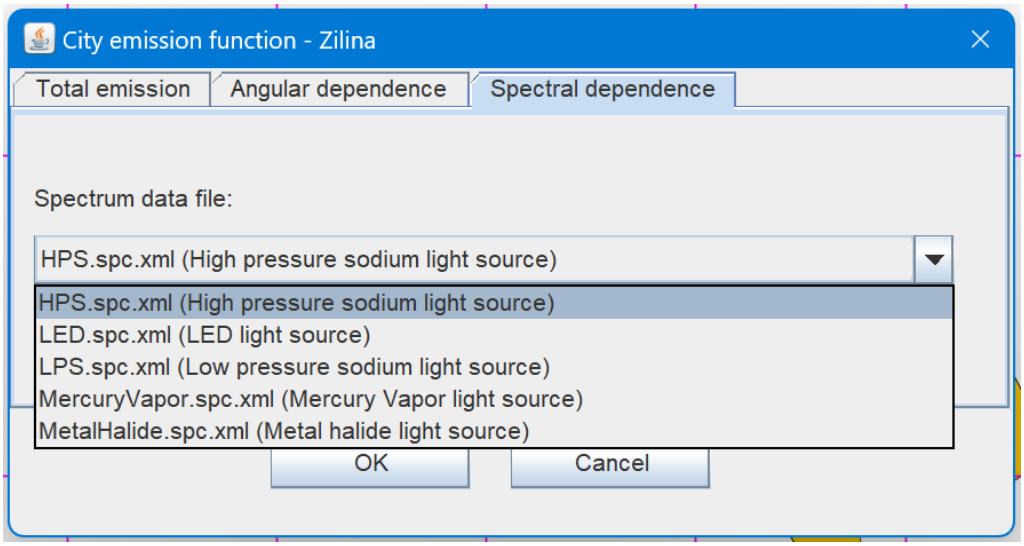


In the "Total emmision" tab you can specify the total emission of the city in Wattts or in lumens. The program automatically recalculates Watts to lumens (or back) using known spectrum of the light sources and using human-eye scotopic sensitivity curve. Eventually, you can estimate the emission using the city population and the typical number of lumens per habitant in your locality.

In the "Angular dependence" tab you can specify the angular pattern of the city emission function. We use two-parametric axisymmetric Garstang's function. Two parameters that can be specified are: surface isotropic reflectivity and the fraction of light directly radiated toward the sky. Set this second parameter to zero and the city will be modelled as a cosine-source. Current version of the program does not support angular dependence in the form of a data-table.

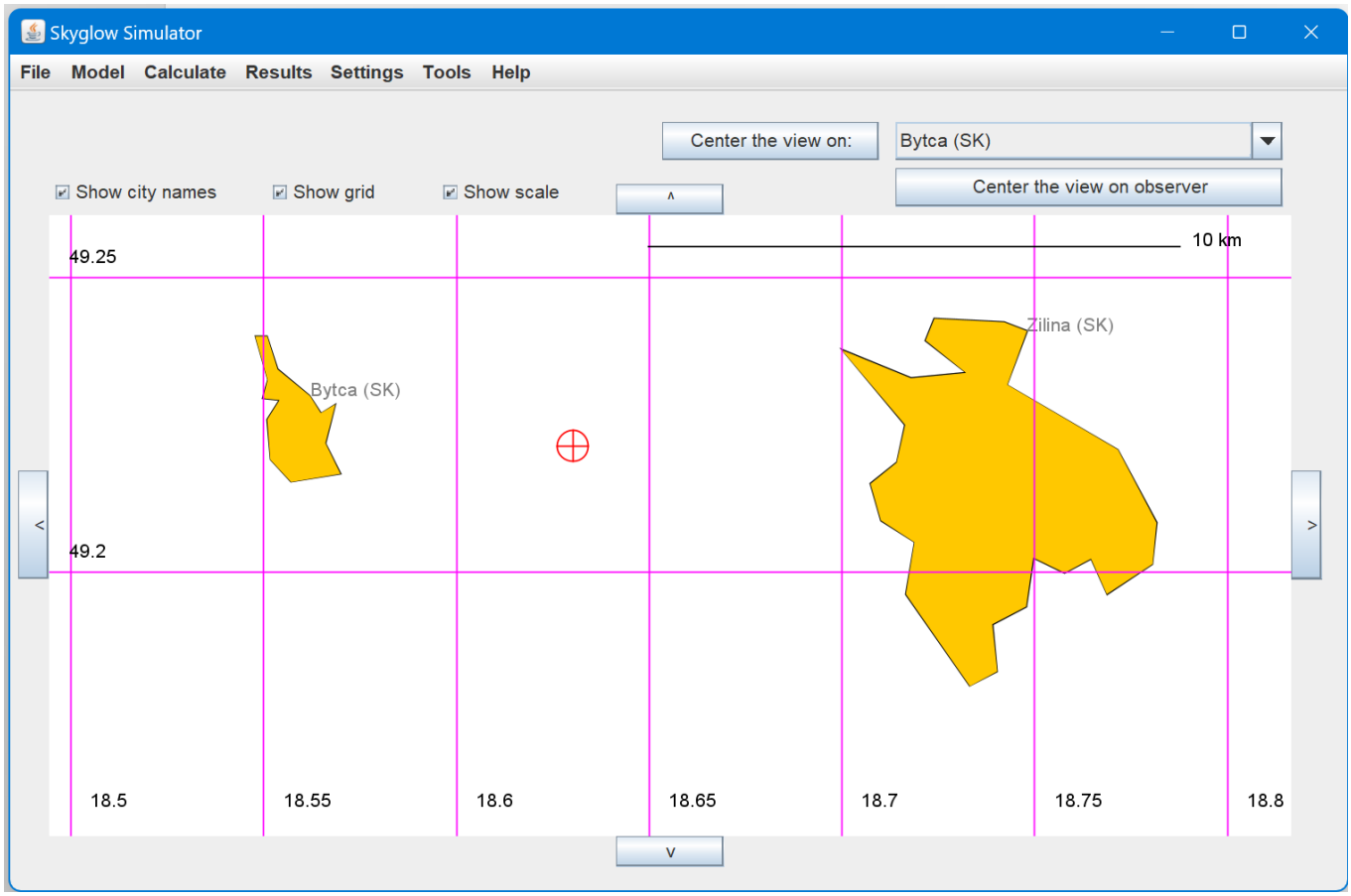


In the "Spectral dependence" tab you can select one typical spectra from the prepared set.



Setting the position of the observer

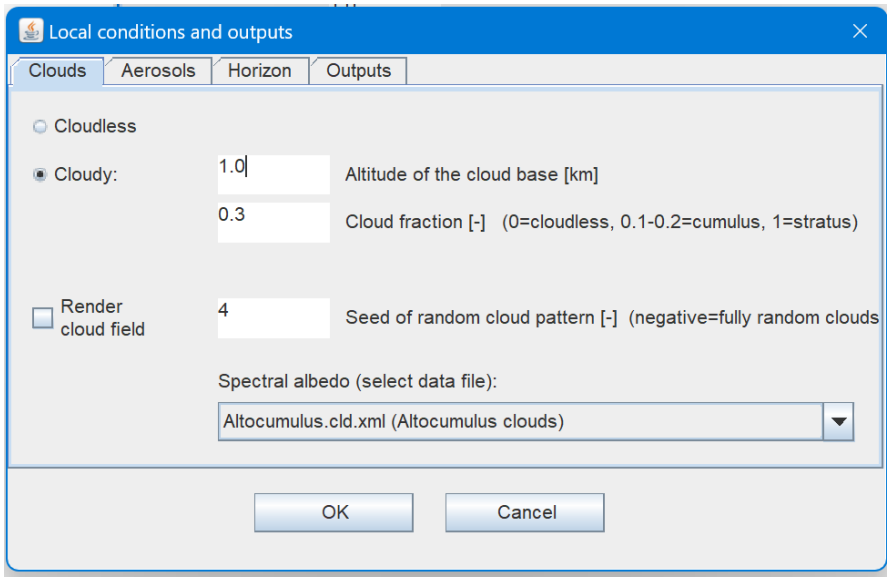
To set the position of the observer **left-double-click** on any place in the map, e.g.:



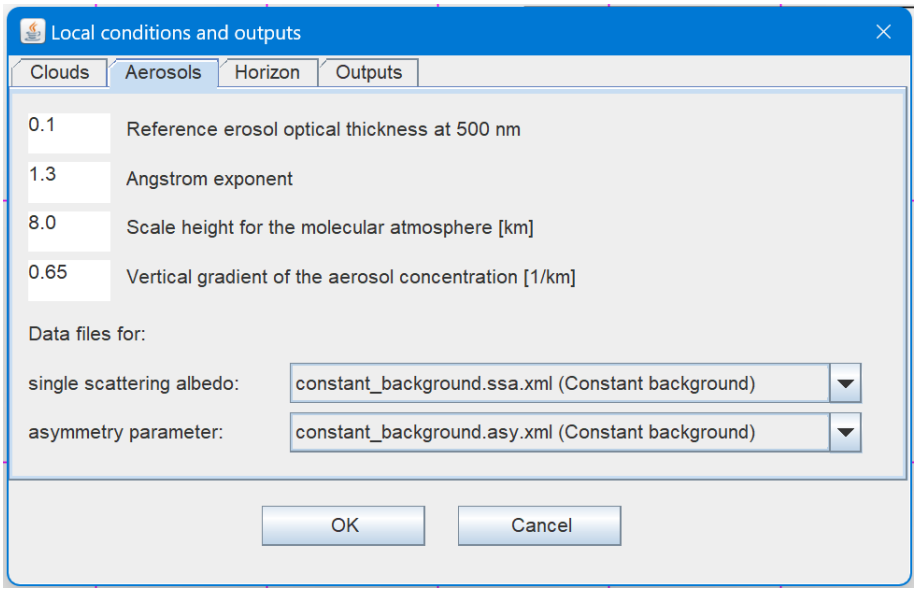
If the exact GPS position of the observer is known, enter its **latitude and longitude** using "**Model/Set observer's position**" menu item.

Changing the local parameters (atmosphere, horizon) and specifying the outputs

To change the local parameters **right-click** the map **outside any city**. The dialog will be displayed.

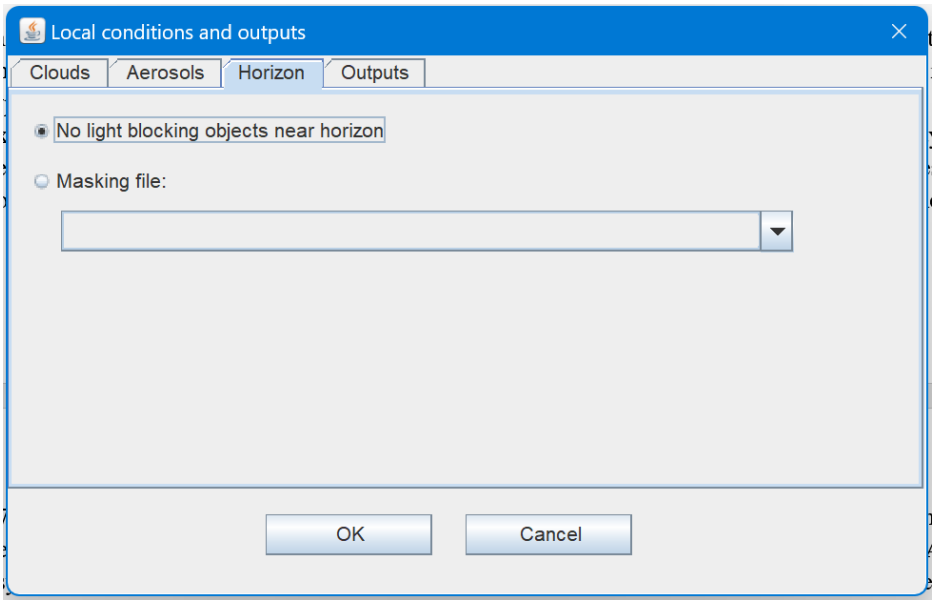


In the "Clouds" tab you can specify cloudless or cloudy sky. The altitude of the cloud base in kilometers can be specified. The cloud fraction of 1 means totally cloudy sky. The cloud fraction less than 1 specifies partly cloudy condition. As an example, we can set the cloud fraction to 0.3. Clouds are generated at random position and size, the seed of random generator can be specified (seed is set to 4 as an example). Negative value of the seed means that the seed is determined by the computer time, so the cloud pattern differs in any calculation. You can check the "Render cloud field" checkbox to generate pictures of some views to the cloud layer. Finally, the spectral reflectance of clouds can be selected from prepared set.

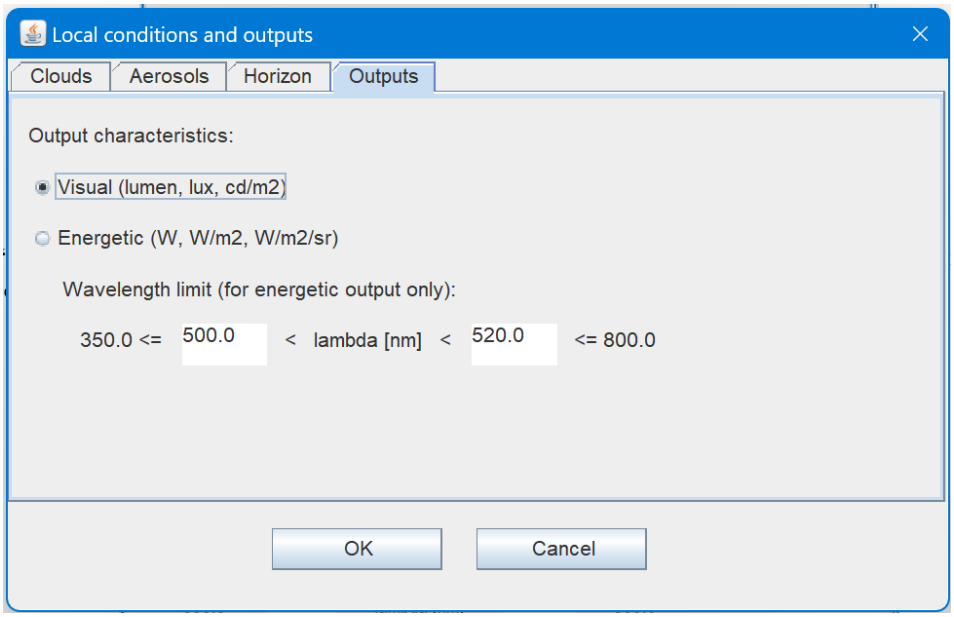


In the "Aerosols" tab you can specify the parameters of aerosols. Reference aerosol thickness determines the amount of aerosol. Optical thickness less than 0.1 means very clear atmosphere, the value of 0.2 represents the typical not very turbid atmosphere, the value about 0.4 means very turbid (smoggy) atmosphere. Angstrom exponent determines the spectral dependence of the aerosol optical thickness. It depends on the type of the aerosol and its typical value is 1.1 – 1.5. Vertical gradient of the aerosol concentration determines how fast the concentration decreases with the height. The number represents the height where the aerosol concentration falls to

37% of its value at the Earth's surface. Similar parameter is the scale height of the molecular atmosphere and its very common value is 8 km. Finally, the spectral dependence of the single scattering albedo (SSA) and of the asymmetry parameter (ASY) is to be selected from prepared files. All the aerosol parameters needed in this tab are measured in AERONET-network stations located around the globe.



In the "Horizon" tab you can eventually select the horizon-profile at the observer's position (dependence of the elevation on the azimuth).

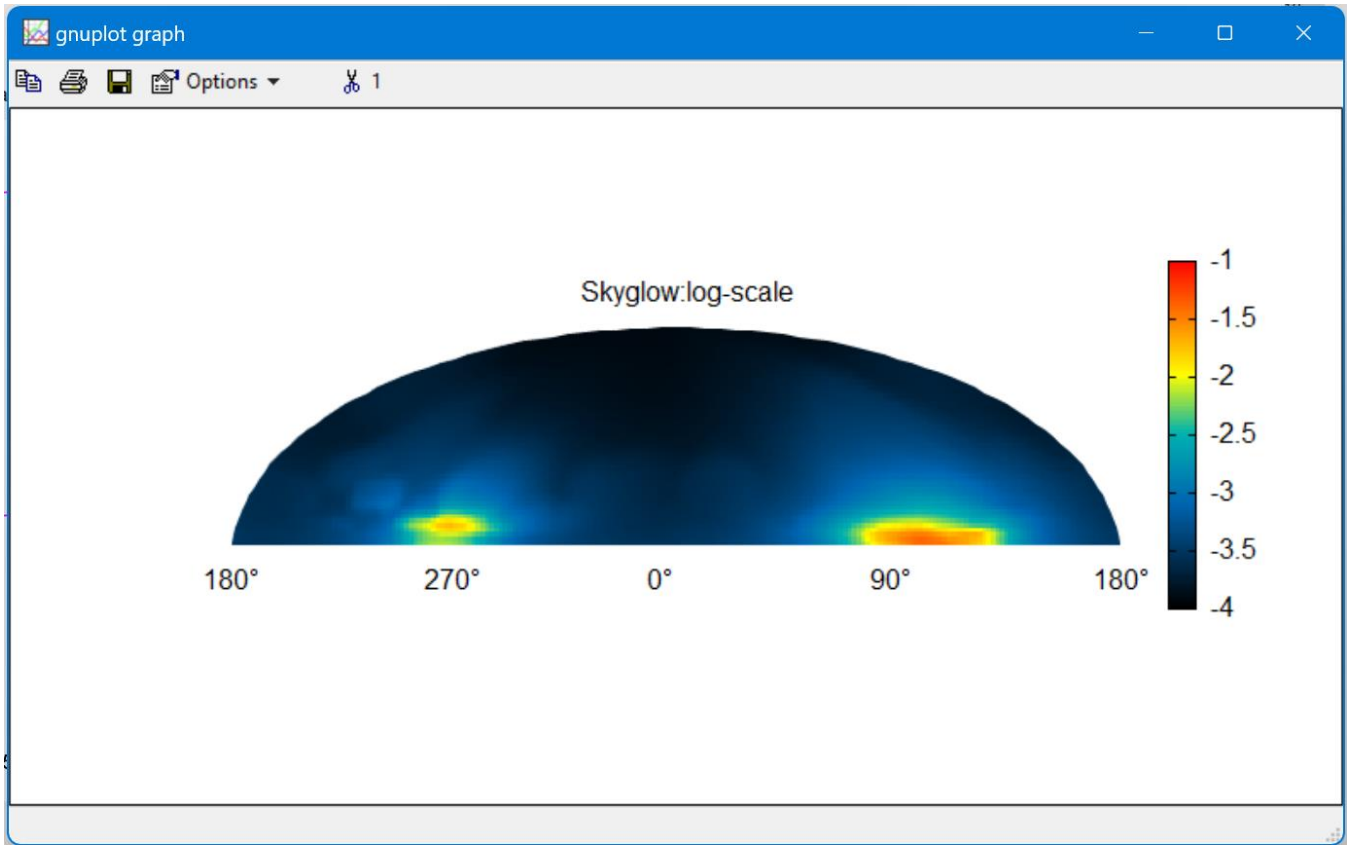


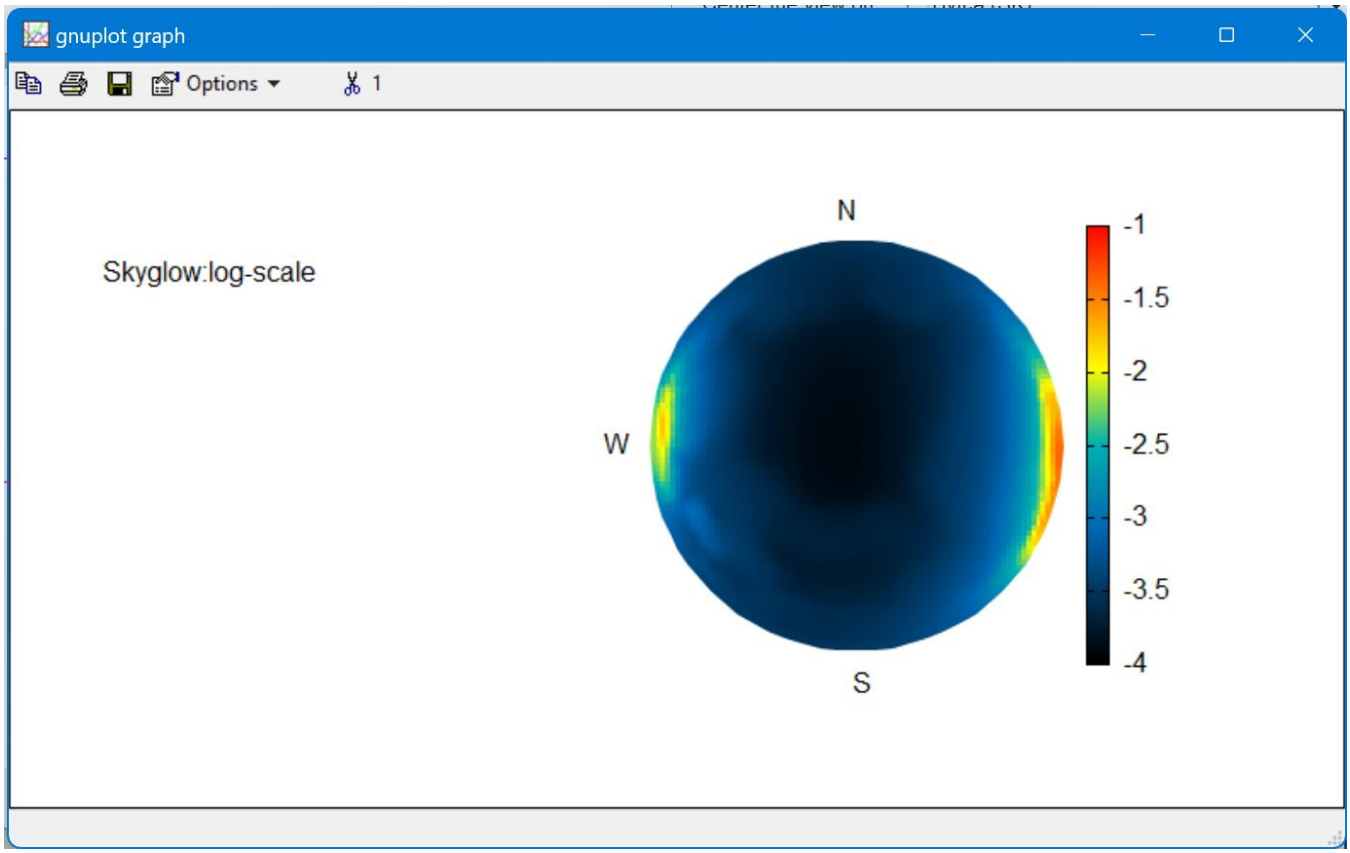
In the "Outputs" tab the output units can be specified. If "Visual" is selected (as is in the example), the skyglow in cd/m^2 is calculated. If "Energetic" is selected, the skyglow in $W/m^2/sr$ is calculated within the selected wavelength interval. By default, the wavelength-step of 5 nm is used in the program. If narrower interval is set, the skyglow is calculated at ends of the interval.

Alternatively, the local parameters can be set using "**Model/Clouds…**", "**Model/Aerosols…**", "**Model/Horizon…**" and "**Calculate/Set output type…**" menu items.

Displaying the results

After the solver is finished (use “Calculate/Save project & Solve” menu item), our demo project has following results (in cd/m^2, logarithm):





In Hammer projection you can specify the direction of observer's view (the azimuth of the center of the plot) in the "Results/Set observer's direction of view" menu item.

The appearance of the plots can be easily modified in corresponding PLT – files located in "results" folder. These files contain GNUplot commands and all commands are commented (explained). As an example, the section that defines the color palette looks like:

```
#--- following parameters significantly influence the colors of the plot --------
#--- uncomment (remove the # character at the beginning of the row) one of the pre-defined palettes ---
#
#set palette rgbformulae 7,5,15          #default color palette (black-blue-red-yellow)
#set palette rgbformulae 8,9,10         #color palette for B&W printing (blue-yellow)
#set palette rgbformulae 30,31,32       #another color palette for B&W printing (black-blue-magenta-
yellow-white)
#set palette defined (0 "black", 2 "blue", 4 "yellow", 6 "red") #color palette (black-blue-yellow-red)
set palette defined (0 0 0 0, 1.8 0 0.4 0.7, 3 0 0.7 0.7 , 4 1 1 0, 6 1 0 0) #color palette (black-cyan-
green-yellow-red)
#set palette gray negative              #inverted greyscale palette
#set palette gray                       #greyscale palette
```

By commenting/uncommenting the line the user can change the color palette. In another sections of the PLT – files user can: set fixed scale of the plots (cbrange), set/unset the isolines (cntrparam), set the font size and picture resolution (set terminal), change the size and position of the legend (colorbox), change numbers in the legend (clabel), set the text and the position of the title (set key), add any text at any position (set label), …

If the "global" PLT-files in "programs/windows" folder are modified, the change will be applied on all newly created pictures.

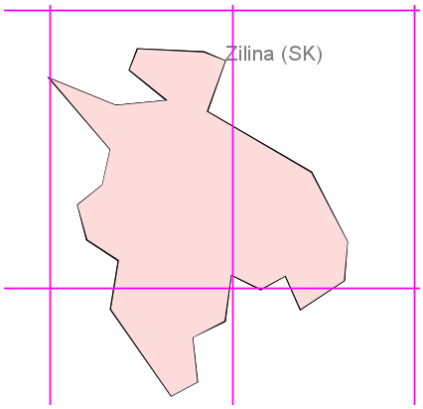
Use "**Help/Plots formatting**" menu item for more information.

Splitting/enlarging the city

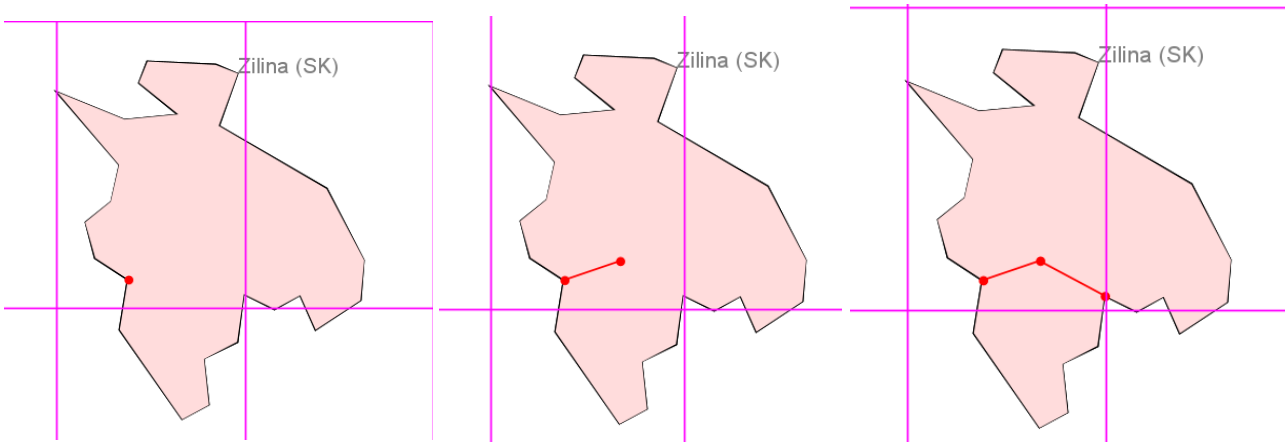
If you want to change parameters of any part of the city separately, we strongly recommend to prepare the parts of the city as separate city-files. However, some basic splitting/enlarging of the city is possible directly in the GUI. In any case, create a backup of the city-files before changing them in GUI.

How to split the city into two parts

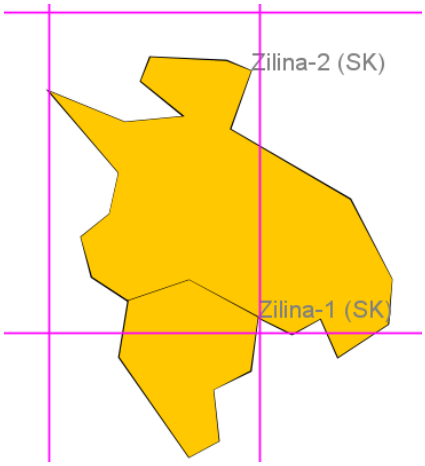
Select "**Tools/Enlarge/split a city (1 new vertex)**" menu item and a brief help will be displayed. Then **left-click** the city you want to split. The city changes the color to pink:



Then, select the first vertex of the cutting-line, e.g.:

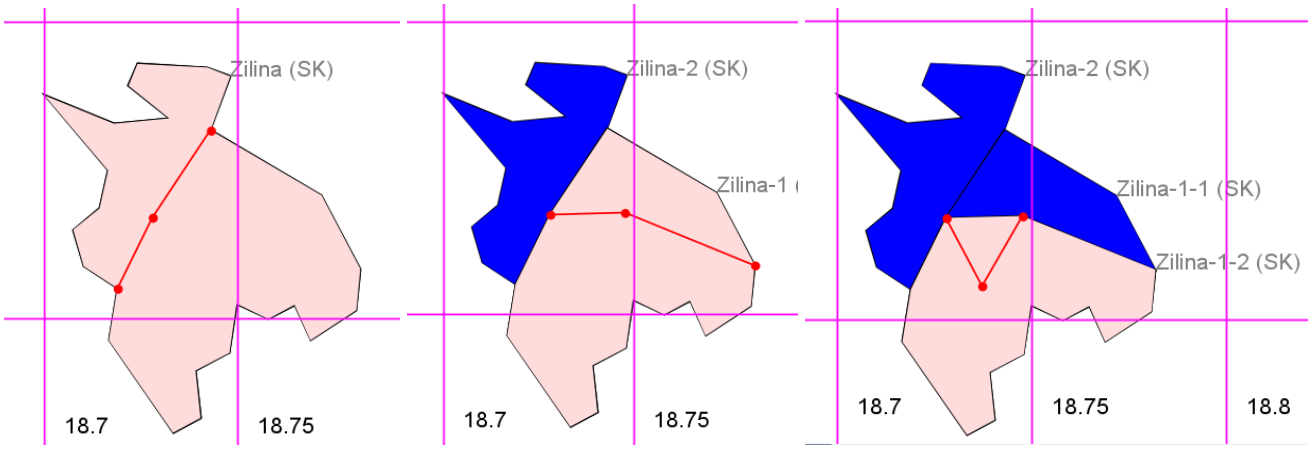


Then click any **point inside the city** and finally, another vertex of the city. In any step, right-click removes the last action. If you are happy with the cutting-line, double click the city:

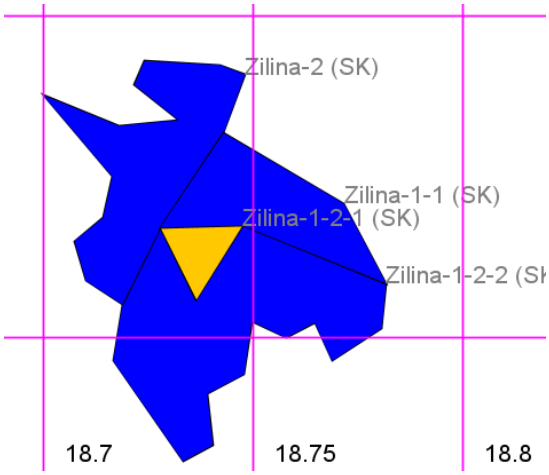


As you can see, the city “Zilina” is now split into 2 parts: “Zilina-1” and “Zilina-2”. The GUI automatically adjusted the total emission and/or the population of both parts according to their area. Now, the city emission function (total emission, angular dependence, spectrum) of any part can be set independently.

Although the GUI is only intended for simple slicing of the city, it is possible to carve out the inner parts of the city by repeating the splitting-process. As an example, we will separate a triangular area. The procedure is shown step by step in the next figure:



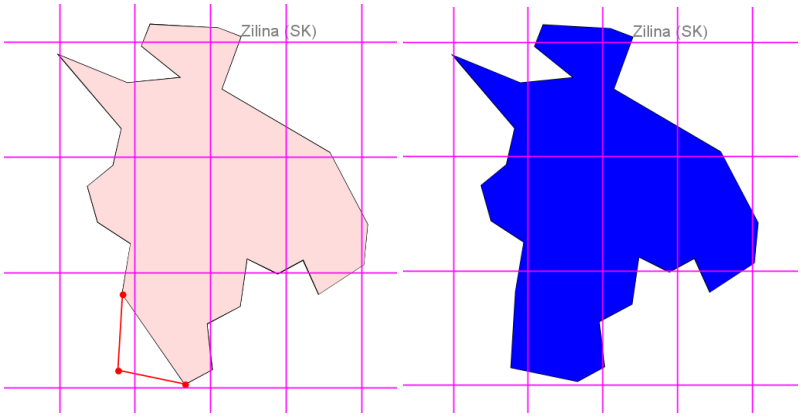
As you can see, the triangular area inside the city can be selected separately now:



Of course, analogically you can separate more complicated polygons.

How to enlarge the city

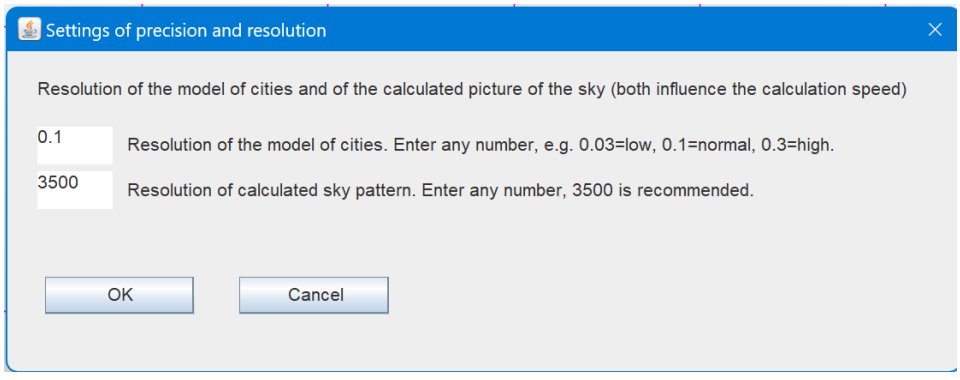
Select "**Enlarge/split a city (1 new vertex)**" radio button and **left-click** the city you want to split. The city changes the color to pink. Then, select the first vertex of the new border-line. Then click any **point outside the city** and finally, another vertex of the city. In any step, right-click removes the last action. If you are happy with the new border-line, double click the city:



The GUI automatically adjusted the total emission and/or the population according to the increased city area.

Setting the precision/resolution

Select the "Calculate/Set resolution…" menu item – a dialog will be displayed:



The first parameter defines the density of points sources placed inside the city polygons. The higher is the density, the smoother and more accurate is the calculated skyglow pattern. The point sources used in the calculation are shown in “city polygon *.png” file(s) stored in the “results” folder.

The second parameter represents the number of points in the hemisphere where the skyglow is calculated. This parameter controls the resolution of the calculated output only, it does not influence the precision of the calculated values.

Both parameters significantly influence the calculation time.

6. Creating own model

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. The data in *.cty.xml files are used for geographical latitude and longitude of all vertices that define the polygonal map of a city. At present, only two city emission functions (cefs) can be used: Garstang’s (includes surface reflectivity and fraction of light emitted directly upwards) and cosine (zero direct uplight in Garstang’s function). Data-file input is currently under development. Spectra of light sources can be defined in lights*.spc.xml files (the number of data is limited to 200). Spectral profiles of aerosol asymmetry parameter (*.asy.xml) and single scattering albedo (*.ssa.xml) 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.xml). The obstacles are simulated through horizon shielding – see the files horizons*.msk.xml where the <key> is the azimuth while the <value> 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 or partly-cloudy conditions (many layers of the atmosphere have to be considered) and/or the photometric output (many wavelengths have to be calculated) can lead to more time-expensive calculations.

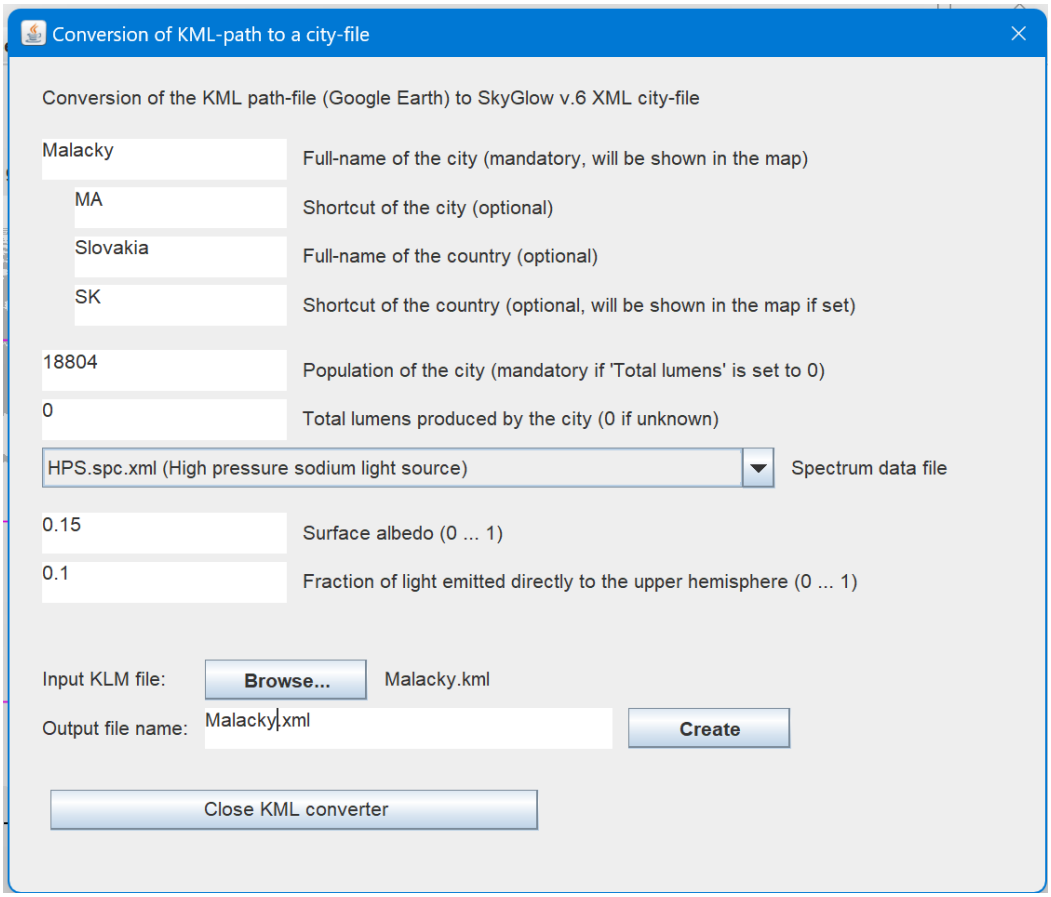
Note 3: At present, only one city emission functions (CEFs) can be used (Garstang). Data-input is under development.

Some hints how to prepare required data-files are collected below.

Creating the city-files

In the SkyGlow program, cities are modeled as surface light sources bounded by a polygon. A polygon is defined by the location (latitude and longitude) of its vertices. A simple way to obtain the bounding polygon of the city is to use the **Google Earth application**, which allows you to mark the so-called "path" and save it (including breakpoints) to a KML file in structured format. The KML file can be then converted into the XML format used by SkyGlow (city polygon vertices + metadata) in simple steps.
Note, that very detailed city boundary do not result in more accurate sky pattern. A few tenths of points is more than enough.

City digitization procedure:
1. Start the Google Earth program and zoom in on the city (for example, Malacky).
2. Select "Path" in the “Add”-menu, enter its name (for example "Malacky").
3. Use the mouse to mark the edge points of the city (a line is drawn with the vertices highlighted). The path need not to be closed, it will be closed automatically in later steps.
4. Click “OK” when finished.
5. Save the marked path (place) in KML format (for example as Malacky.kml) – don’t use compressed KMZ format!
Comment: In **Google Earth WEB interface**, the process is very similar: create a polygon and save it locally. Then export the polygon as a KLM file. Next steps (conversion) are identical.
6. Open Tools/Convert KML-file into city-file menu item:



7. Fill-in all known information about the city. If ‘total lumens’ produced by the city is unknown, it will be estimated from the city population (typical value is 750 lm per habitant and this value can be changed later in the SkyGlow GUI). Typical value of surface albedo is 0.1-0.2. Cities with older types of luminairs have typical uplight 0.1-0.2. Modern LED luminairs have close-to-zero uplight. However, windows and monumental lighting produce uplight with typical value of 0.05-0.1 of total lumens produced by the city.
8. Click “Browse” button and select the KML-file (e.g. ‘Malacky.kml’).
9. Fill in the name of the city file (its extension must be .xml), e.g. ‘Malacky.xml’.
10. Press “Create” button. The city-file will be created and stored in ‘cities’-folder.
11. Repeat steps 6 – 10 for other KML-files.
12. When finished, press “Close KML converter” button.
Comment: To see newly created cities in SkyGlow application, close it and start again.

Creating the spectra of lights

The spectra of common light sources are available online or are distributed by the producers of light sources. The density of wavelength values in such spectra is usually not constant – higher density is near peaks. As SkyGlow calculates the luminance at wavelengths with 5 nm step, it is easy to skip some peaks. So, beside the change to XML format, the aim of the conversion is to create the spectrum with 5 nm step and with integrated values within the 5 nm intervals.

The spectrum is calculated from 350 nm to 800 nm to cover all visible spectrum. The values outside this interval are omitted. If the wavelength interval in source file is narrower, ends of the calculated spectrum are filled with zeros.

The conversion process is as follows:

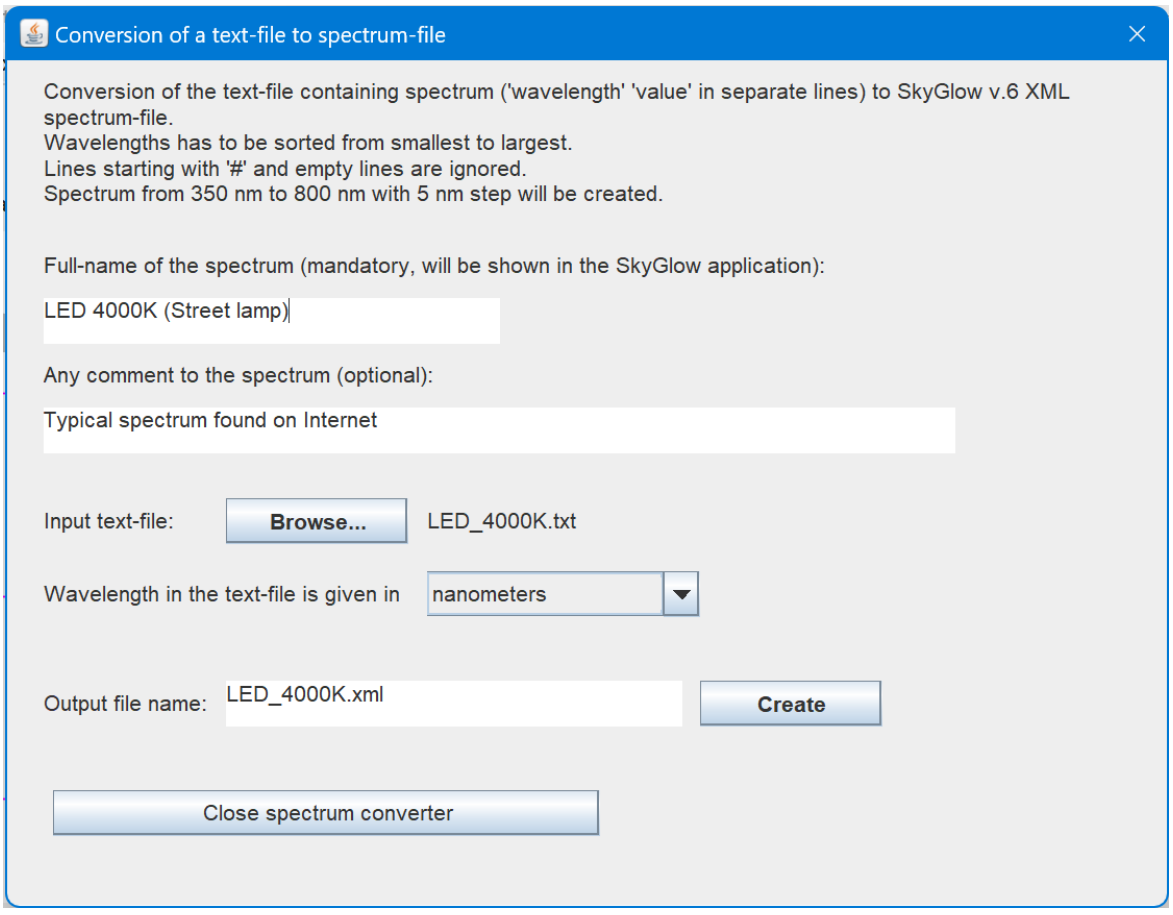
1. Prepare spectrum in form of simple text file with wavelengths sorted in ascending order. Empty lines and lines starting with '#' are ignored. The values in a line must by separated by space or by tabulators (don't use comma separated values). Wavelengths can be given in nanometers, angstroms or micrometers. A comma or a dot can be used as decimal separator.

   Example of an input file:

```
# 4000K LED spectrum
# wavelength[nm] value[-]
345 0.00190
350   0.00214
353   0.00232
355   0.00247
359   0.00289
...
```

   Save this file as the text-file, "LED_4000K.txt" as an example.

2. Open 'Tools/Create spectrum-file from text-file' menu item:

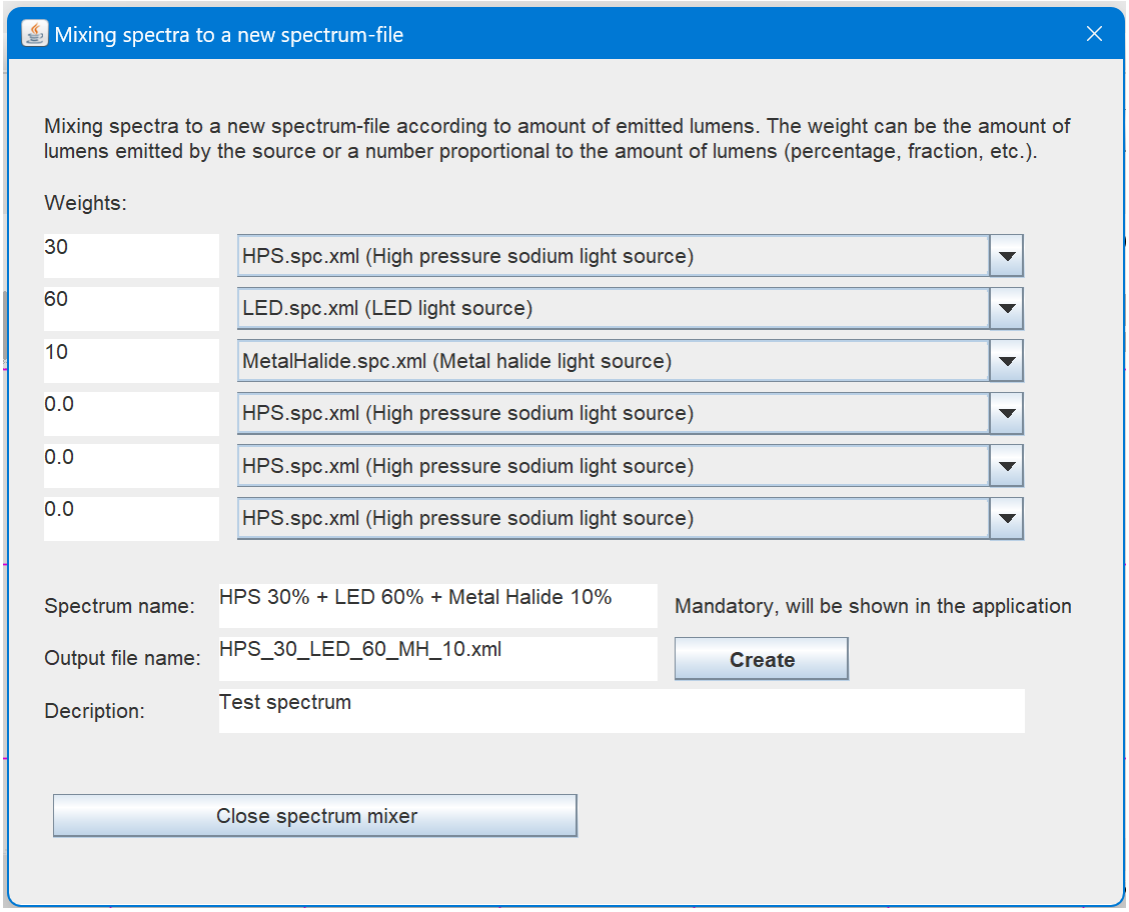


3. Fill-in the name of the spectrum and optionally any comment.
4. Browse for input text file (e.g. ‘LED_4000K.txt’).
5. Select in the combo-box the unit used in the input-file for wavelengths.
6. Fill-in the name of the output file (its extension must be ‘.xml’), e.g. ‘LED_4000K.xml’.
6. Press ‘Create’ button.

Repeat steps 3-6 for other prepared text-files containing spectra. Press ‘Close spectrum converter’ when finished.
Comment: The newly created spectra are stored in ‘lights’-folder. After restarting the SkyGlow application they will be selectable in dialogs.

Mixing the spectra of lights

The spectrum of a city is a mixture of spectra of individual light types (HPS, LED, incandescent, …). If you know the amount of lumens produced by individual light types (or the percentage), you can create a mixed spectrum using ‘Tools/Mix spectra into a new spectrum’ menu item:



1. Select the light sources and fill in corresponding amounts of lumens (or percentage of lumens).
2. Fill in the name of the spectrum, the filename of the new spectrum-file and (optionally) the description of the new spectrum.
3. Press ‘Create’ button.

Repeat previous steps if another mixed spectrum is needed. Press ‘Close spectrum mixer’ button to finish.
Comment: The newly created spectrum will be available in the SkyGlow application after its restart.

Creating the spectral reflectivity of clouds

The cloud '.xml' files are self-explanatory. <key> contains the wavelength and <value> contains the reflectivity.

Creating the horizon profiles

The horizon '.xml' files are self-explanatory. <key> contains the azimuth and <value> contains the elevation.

Creating the aerosol SSA and ASY files

The SSA '.xml and ASY '.xml' files are self-explanatory. <key> contains the wavelength and <value> contains the SSA/ASYvalue. The typical spectral dependencies of SSA/ASY for aerosol at given location can be found in data from AERONET-stations. See link: https://aeronet.gsfc.nasa.gov/cgi-bin/draw_map_display_inv_v3
Note that the inverted data are available later (at least 1 year) after the re-calibration of the devices. Besides SSA and ASY data you can find here Aerosol Optical Depth (Extinction) at 500 nm needed by Skyglow software and from the spectral dependence of the AOD you can also obtain the Angstrom exponent.

7. 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 GUI outputs, such as information on your settings and actions performed before solver (or viewer) was launched. Error-messages are also stored here. Look into 'logfile.txt', if program crashes or some city is not displayed on the map and so on.

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.