

# Satellite-Radar SIMulator (SR-SIM)

## User's Guide

Version 1  
Bernat Puigdomènech Treserras  
[bernat.ptreserras@mcgill.ca](mailto:bernat.ptreserras@mcgill.ca)  
May, 2017

### 1. Introduction

SR-SIM is a Fortran 90 package that computes synthetic satellite radar observations using output data from the high-resolution Weather Research and Forecasting Model (WRF) and Global Environmental Multi-scale Model (GEM, Côté et al., 1998).

SR-SIM simulates radar reflectivity and attenuation consistent with the specifications of the spaceborne radar and the model microphysics parametrization. The package includes a beam convolution routine to adjust the spatial resolution to a sensor field of view (FOV) and an orbit space interpolation routine, to allow direct comparison between the model output and the actual satellite radar observations.

In the current version, SR-SIM is using the specifications of the GPM Dual-frequency Precipitation Radar (DPR), consisting of a Ku-band and a Ka-band radars with frequencies of 13.6 GHz and 35.5 GHz respectively. However, observations from many other satellite sensors can also be simulated, such as EarthCare, CloudSat, Cloud Profiling Radar, etc. taking into account their orbit and spatial resolution characteristics. Therefore, the SR-SIM is suitable for current and for future satellite-borne radars.

The Milbrant-Yau double-moment bulk microphysics parametrization is the microphysics scheme used in the present version, Milbrant et Yau (2005a) and (2005b). SR-SIM was designed to allow for easily incorporating new code, ingesting of new input model or satellite files, modifying the particle size distribution (PSD) settings, etc.

Particular attention was devoted to making the model run efficiently, using lookup tables with pre-computed results. It also runs using shared memory parallel processing (OpenMP) to accelerate the computations.

### 2. Installation

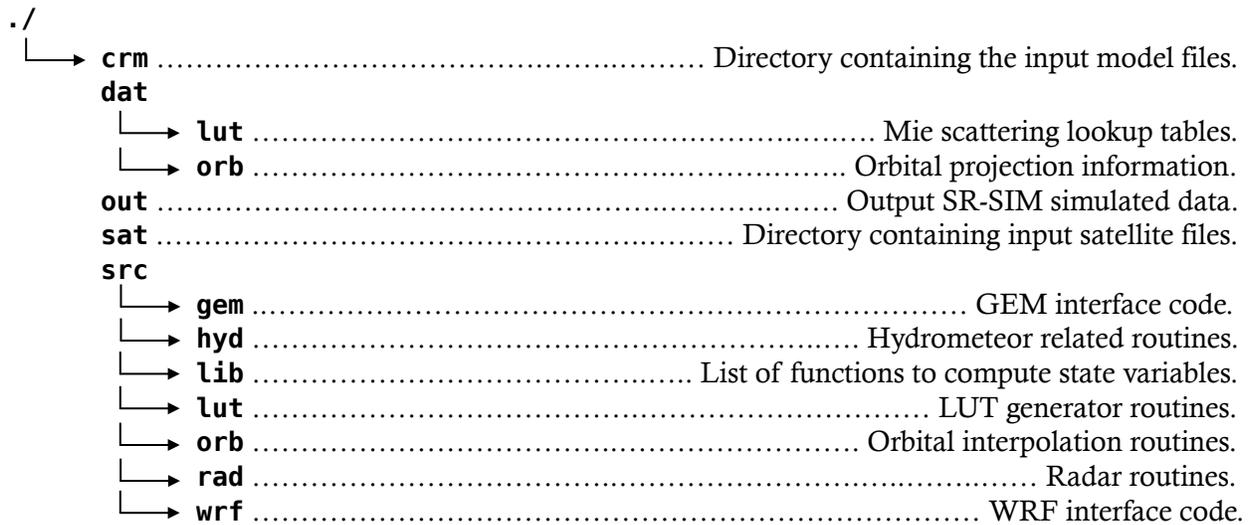
The SR-SIM package is available for download from the following web site:

<http://meteo.mcgill.ca/~bernat/sr-sim/>

Move the file in the directory where you would like to install the package and then unzip and expand the tar ball file:

```
tar -xzvf sr-sim.v1.tar.gz
```

The package consists of the following directories:



In the source tree `src/`, there are 2 executable scripts, `srsim.x` and `make_lut.x`. The first one is the main simulator unit and the second one the Mie scattering LUT generator. Input model files are read from `crm/` and satellite files from `sat/` (optional).

These 2 folders can be omitted when the main simulator program is called by passing absolute file paths. Note that the downloaded compiled version is ready to be executed and contains an example of a WRF simulation and an actual GPM file for comparison

The output synthetic observations are written out into `out/`.

For more information about configuration and execution, refer to the section 5.

### 3. Compilation

From the directory `src/` you can edit the `makefile`, specifying the compiler installed on your computer. Note that, by default, the compiler used is `ifort` (Intel Fortran Compiler). You can change this modifying the content of the flag `COMPL` and the compiling options in `FLAGS_N` and `OPENMP`. This last flag, set to `(yes/no)` specifies the use of parallel processing using.

Once the makefile is properly setup, you can run it:

```
make
```

If no errors occur, the executable file `srsim.x` will be created.

The `clean` option will remove all object and module intermediate files, required for the creation of the executable file:

```
make clean
```

Make use of the makefile every time you change any general setting.

## 4. General Settings

The settings that can be modified by the user are defined in the module file `srsim_param.f90`. Re-compilation is required after modifying any of these options.

```
module srsim_param

!> Satellite radar name
  character*(*), parameter :: radar_name = 'GPM Core Observatory'

!> Input CRM name, 2 types supported (GEM/WRF)
  character*(*), parameter :: crm_type = 'WRF'

!> Micrphysical scheme. The corresponding routine must be defined in psd_hydro.f90
  character*(*), parameter :: phys = 'MilbYau'

!> Save Reflectivity and PIA for each hydrometeor category
  logical, parameter :: save_hydro = .true.

!> Number of channel frequencies
  integer, parameter :: nF = 2

!> Radar channel frequencies Ku-band (13.6GHz) and Ka-band (35.5GHz)
  real, dimension(nF), parameter :: Freq_rad = (/ 13.6, 35.5 /)

!> Minimum detectable echo
  real, dimension(nF), parameter :: min_echo = (/ 13.5, 12.5 /)

!> Configuration for the radar antenna beam convolution, sensor field of view in [km]
  real, dimension(nF), parameter :: fov_ct_radar = (/ 5.2, 5.2 /)
  real, dimension(nF), parameter :: fov_dt_radar = (/ 5.2, 5.2 /)

!> Viewing zenith angle [deg]
  real, parameter :: zenith_angle = 0.

!> Generate melting particles
  logical, parameter :: meltp = .true.

!> Radar antenna beam convolution
  logical, parameter :: beam_conv = .true.

!> Satellite orbit projection
  logical, parameter :: orbit_proj = .true.

!> Print informative messages during the program execution
  logical, parameter :: verbose = .true.

!> Define missing data value
  real :: nodata = -9999.

!> Cloud Resolving Model file. Input from command line
  character*150 :: CRM_file

!> GPM. Input from command line
  character*150 :: SAT_file

!> Input SAT name, 1 type supported (GPM)
  character*(*), parameter :: sat_type = 'GPM'

!> GPM scan type (MS/NS/HS)
  character*(*), parameter :: gpm_scan = "NS"

!> LUT directory
```

```

    character*(*), parameter :: lut_dir = "../dat/lut/"
!> Orbit directory
    character*(*), parameter :: orb_dir = "../dat/orb/"
!> Output directory
    character*(*), parameter :: out_dir = "../out/"
end module srsim_param

```

## 5. Particle Size Distribution (PSD) library

SR-SIM is currently using the PSD model described in Milbrant et Yau (2005a) and (2005b) in the form of a Fortran function defined in the file **psd\_hydro.f90**. The user can add new PSD models following the specifications of that file (see the code documentation for more details).

In order to use a specific PSD model, the parameter **phys** in **srsim\_param.f90** must be properly set to the name of the model.

After a new model is introduced to SR-SIM, the code must be re-compiled and the lookup tables of the hydrometeor radiative properties must be generated prior to execution of the main program.

## 6. Mie scattering Lookup Tables (LUT)

SR-SIM makes use of lookup tables containing the hydrometeor radiative properties to speed up computations. The LUT must be created in advance off-line before execution. The routines that manage the configuration and generation of LUT are located under the directory **src/lut**.

### 6.1 Compilation of the LUT generator

The LUT generator can be compiled making use of the **makefile** under **src/lut/**. The compiler options and flags are exactly the same as the simulator unit (see section 3).

Once the makefile is properly setup, you can run it:

```
make
```

If no errors occurred, the executable file **make\_lut.x** will be created in the **src** level of the distribution.

The **clean** option will remove all object and module intermediate files, required for the creation of the executable file:

```
make clean
```

Finally, the **clean\_lut** option will remove all LUT tables created in the LUT directory (set to **dat** by default).

## 6.2. LUT generator options

The settings of the LUT generator can be modified in the module file **lut\_param.f90**. Re-compilation is required after modifying any of these options.

```
module lut_param

!> Micrphysical scheme. The corresponding routine must be defined in psd_hydro.f90
  character*(*), parameter :: phys = 'MilbYau'

!> Number of channel frequencies
  integer, parameter :: nF = 2

!> Radar channel frequencies. GPM uses Ku-band (13.6GHz) and Ka-band (35.5GHz)
  real, dimension(nF), parameter :: Freq_rad = (/ 13.6, 35.5 /)

!> LUT directory
  character*(*), parameter :: LUT_dir = ".././dat/lut/"

!> W sampling points
  integer, parameter :: nW = 39

!> Sampling liquid water content [g/m3]
  real, dimension(nW), parameter :: Wp_sampl = (/ 0.00025, 0.0005, 0.001, 0.002, &
    0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, &
    0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, &
    1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0 /)

!> T sampling points
  integer, parameter :: nT = 15

!> Sampling temperature [K] for liquid hydrometeors (rain and cloud water)
  real, dimension(nT), parameter :: Tw_sampl = (/ 245.0, 250.0, 255.0, 260.0, &
    265.0, 270.0, 275.0, 280.0, 285.0, 290.0, 295.0, 300.0, 305.0, 310.0, 315.0 /)

!> Sampling temperature [K] for frozen hydrometeors (cloud ice, snow, graupel, and
hail)
  real, dimension(nT), parameter :: Ti_sampl = (/ 180.0, 190.0, 200.0, 210.0, &
    220.0, 230.0, 240.0, 250.0, 260.0, 270.0, 275.0, 280.0, 285.0, 290.0, 295.0 /)

!> N sampling points
  integer, parameter :: nN = 31

!> Sampling number concentration sampling vector for cloud particles [1/m3]
  real, dimension(nN), parameter :: Nc_sampl = (/ 0.0, 1.0e+3, 3.0e+3, 1.0e+4, &
    2.0e+4, 4.0e+4, 6.0e+4, 8.0e+4, 1.0e+5, 2.0e+5, 4.0e+5, 6.0e+5, 8.0e+5, &
    1.0e+6, 2.0e+6, 4.0e+6, 6.0e+6, 8.0e+6, 1.0e+7, 2.0e+7, 4.0e+7, 6.0e+7, &
    8.0e+7, 1.0e+8, 2.0e+8, 4.0e+8, 6.0e+8, 8.0e+8, 1.0e+9, 3.0e+9, 1.0e+10 /)

!> Sampling number concentration sampling vector for precipitation particles [1/m3]
  real, dimension(nN), parameter :: Np_sampl = (/ 0.0, 1.0e+0, 3.0e+0, 1.0e+1, &
    2.0e+1, 4.0e+1, 6.0e+1, 8.0e+1, 1.0e+2, 2.0e+2, 4.0e+2, 6.0e+2, 8.0e+2, &
    1.0e+3, 2.0e+3, 4.0e+3, 6.0e+3, 8.0e+3, 1.0e+4, 2.0e+4, 4.0e+4, 6.0e+4, &
    8.0e+4, 1.0e+5, 2.0e+5, 4.0e+5, 6.0e+5, 8.0e+5, 1.0e+6, 3.0e+6, 1.0e+7 /)

!> Print informative messages during the program execution
  logical, parameter :: verbose = .true.

end module lut_param
```

Note that the SR-SIM interpolates between sampling numbers of the LUT. Therefore, more resolution in the sampling arrays may lead to better results.

## 6.3 Run LUT generator

Once all parameters are set, run the Mie scattering LUT generator:

```
./make_lut.x
```

The LUT program will create NetCDF files with all hydrometeor radiative properties. These files will be read by the simulator unit after the ingestion of the model variables.

## 7. Run SR-SIM

To run SR-SIM, go to the directory **src/** and execute **srsim.x** passing the name of the model file you want to use for the simulation. If no other options are specified, SR-SIM will simulate synthetic radar observations for every point of the model grid.

Optionally, you can specify a satellite file in the call. If this is the case, the output simulation will match the orbit and resolution of the satellite observations allowing a direct comparison between the two.

Example:

```
./srsim.x ../crm/wrfout_d02_2014-06-11_20:00:00 -sat ../gpm/  
2A.GPM.DPR.V6-20160118.20140611-S171129-E184401.001619.V04A.HDF5
```

## 8. NWP interface

### 8.1 GEM

The GEM interface (**read\_gem.f90**) was created to work with GEM post-processed NetCDF files. These files contain 3D information defining the model state. The following variables must be present:

```
!> - temperature Temperature [C]  
!> - pres_mass Pressure at the center of the level [hPa]  
!> - humidity Specific Humidity [kg/kg]  
!> - mpqc Cloud mass mixing ratio [kg/kg]  
!> - mpqi Ice mass mixing ratio [kg/kg]  
!> - mpqs Snow mass mixing ratio [kg/kg]  
!> - mpqr Rain mass mixing ratio [kg/kg]  
!> - mpqg Graupel mass mixing ratio [kg/kg]  
!> - mpqh Hail mass mixing ratio [kg/kg]  
!> - mpnc Cloud number mixing ratio [1/kg]  
!> - mpni Ice number mixing ratio [1/kg]  
!> - mpns Snow number mixing ratio [1/kg]  
!> - mpnr Rain number mixing ratio [1/kg]  
!> - mpng Graupel number mixing ratio [1/kg]  
!> - mpnh Hail number mixing ratio [1/kg]  
!> - latitude Latitude [deg]  
!> - longitude Longitude [deg]  
!> - geopot_height Height [decim]
```

With the following dimensions:

```
!> - x
```

```
!> - y
!> - z_mom
```

Note that the grid resolution is meant to be 2.5km.

For more information contact Dominik Jacques ([dominik.jacques@canada.ca](mailto:dominik.jacques@canada.ca)).

## 8.2 WRF

The WRF interface (**read\_wrf.f90**) reads standard WRF model output files. These are NetCDF files containing the following required variables:

```
!> - XLAT Latitude [deg]
!> - XLONG Longitude [deg]
!> - W z-wind component [m/s]
!> - P Perturbation pressure [Pa]
!> - PB Base state pressure [Pa]
!> - T Perturbation potential temperature [K]
!> - PH Perturbation geopotential [m2/s2]
!> - PHB Base state Geopotential [m2/s2]
!> - QVAPOR Water vapor mixing ratio [kg/kg]
!> - QCLOUD Cloud water mixing ratio [kg/kg]
!> - QICE Ice mixing ratio [kg/kg]
!> - QSNOW Snow mixing ratio [kg/kg]
!> - QRAIN Rain mixing ratio [kg/kg]
!> - QGRAUP Graupel mixing ratio [kg/kg]
!> - QHAIL Hail mixing ratio [kg/kg]
!> - QNIC Ice number concentration [1/kg]
!> - QNRN Rain number concentration [1/kg]
!> - QNSN Snow number concentration [1/kg]
!> - QNCLOUD Cloud water number concentration [1/kg]
!> - QNGRAUPEL Graupel number concentration [1/kg]
!> - QNHAIL Hail number concentration [1/kg]
```

With the following dimensions:

```
!> - weast_east
!> - south_north
!> - bottom_top
!> - bottom_top_stag (for staggered variables)
```

The resolution of the grid may be variable but it should be less than the satellite sensor Field of View (FOV).

## 8.1 SR-SIM plug-in library

Note that the variables read from the NWP may need to be converted to the standard units used in SR-SIM. For such purpose, a collection of library functions are included in the SR-SIM package. These routines are included in the directory **lib/**:

```
!> - AirD - Computes dry air density [kg/m3] from temperature [K] and pressure [Pa]
!> - Hght - Computes geopotential height [Km] from perturbation geopotential [m2/s2]
and base-state perturbation geopotential [m2/s2]
!> - Pres - Computes pressure [Pa] from perturbation pressure [Pa] and Base state
pressure [Pa]
!> - Q2Kg - Computes pQ [kg/m3] or pN [1/m3] from any mixing ratio [kg/kg] or number
concentration [1/kg], multiplying by air density [kg/m3]
!> - RLhm - Computes relative humidity [%] from temperature [K], pressure [Pa] and
vapor mixing ratio [kg/kg]
```

!> - **Temp** - Computes temperature [K] from potential temperature [K] and pressure [Pa]  
!> - **Thet** - Computes Potential temperature [K] from perturbation potential temperature [K]  
!> - **WvaP** - Computes water vapor pressure [Pa] from water vapor mixing ratio [kg/kg] and pressure [Pa]

## 9. License and Disclaimer

SR-SIM is based on the Satellite Data Simulator Unit (SDSU-v2) (Masunaga et al., 2010) used under agreement with the original author.

This software is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation v3:

This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

All copies and redistributions of the software must retain the original license and disclaimer notice and the name of the original authors.

See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

## References

Milbrant et Yau (2005), A Multimoment Bulk Microphysics Parameterization. Part I: Analysis of the Role of the Spectral Shape Parameter. DOI: <http://dx.doi.org/10.1175/JAS3534.1>

Milbrant et Yau (2005), A Multimoment Bulk Microphysics Parameterization. Part II: A Proposed Three-Moment Closure and Scheme Description. DOI: <http://dx.doi.org/10.1175/JAS3535.1>

Massunaga et al. (2010), Satellite Data Simulator Unit: A Multisensor, Multispectral Satellite Simulator Package. DOI: <http://dx.doi.org/10.1175/2010BAMS2809.1>