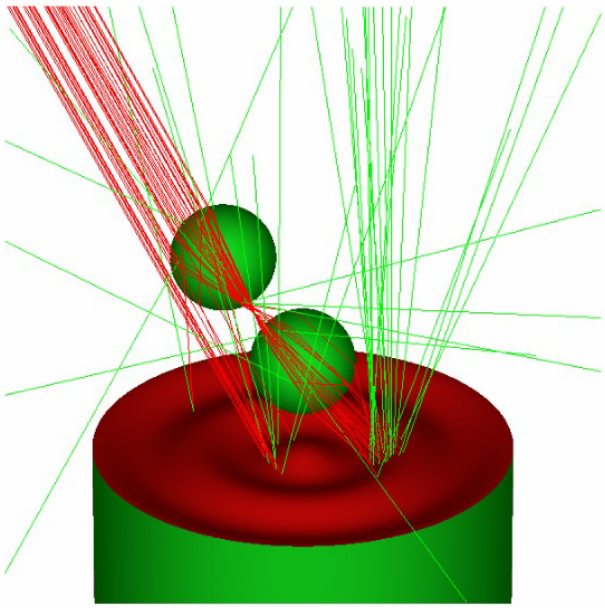
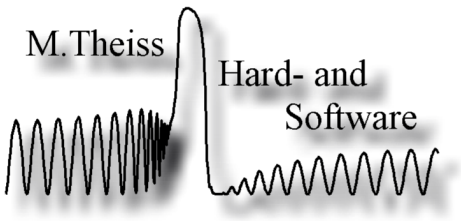
SPRAY

M.Theiss Hard- and Software



Tutorials

M.Theiss Hard- and Software for Optical Spectroscopy
Dr.-Bernhard-Klein-Str. 110, D-52078 Aachen
Phone: (49) 241 5661390 Fax: (49) 241 9529100
E-mail: theiss@mtheiss.com Web: www.mtheiss.com



SPRAY

Spectral ray-tracing simulations

by Wolfgang Theiss

Printed: 01.12.2009, 22:55 in Aachen, Germany

Table of Contents

Part I

1 Introduction

1.1 About this document

Spectral Ray Tracing

Tutorials 1

W. Theiss – Hard- and Software
Dr.-Bernhard-Klein-Str. 110, D-52078 Aachen, Germany
Phone: + 49 241 5661390 Fax: + 49 241 9529100
e-mail: theiss@mtheiss.com web: www.mtheiss.com

August 2008

This set of SPRAY tutorials serves as introduction to the operation of the program. The examples are simple - far from serious scientific or industrial applications.

- Basic skills: Light throughput of a cylinder (fiber)
- Working with light scattering by small particles: Computing diffuse reflectance spectra

Part II

2 Basic skills

2.1 Overview

This tutorial quickly guides you through SPRAY. We will work on a simple example, namely the investigation of the throughput of radiation through a solid cylinder.

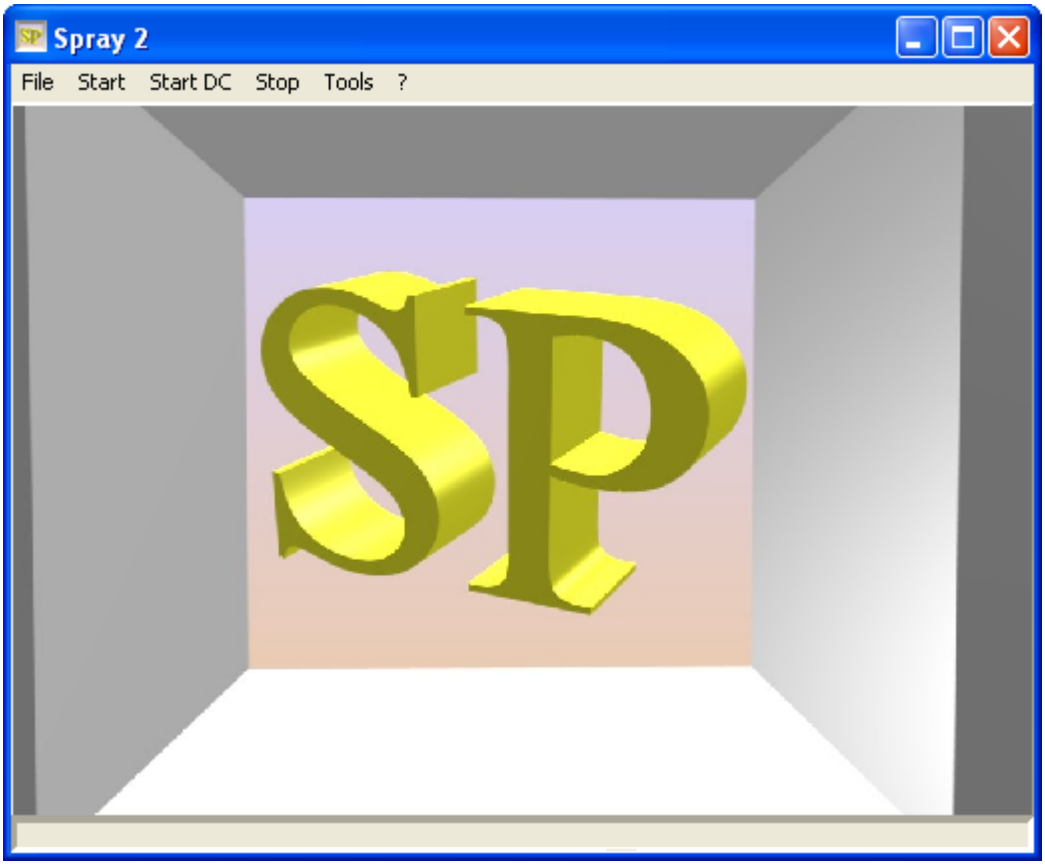
Start SPRAY and try to follow the steps described below. This is the fastest way to learn how to operate the program.

Here are the parts of this tutorial:

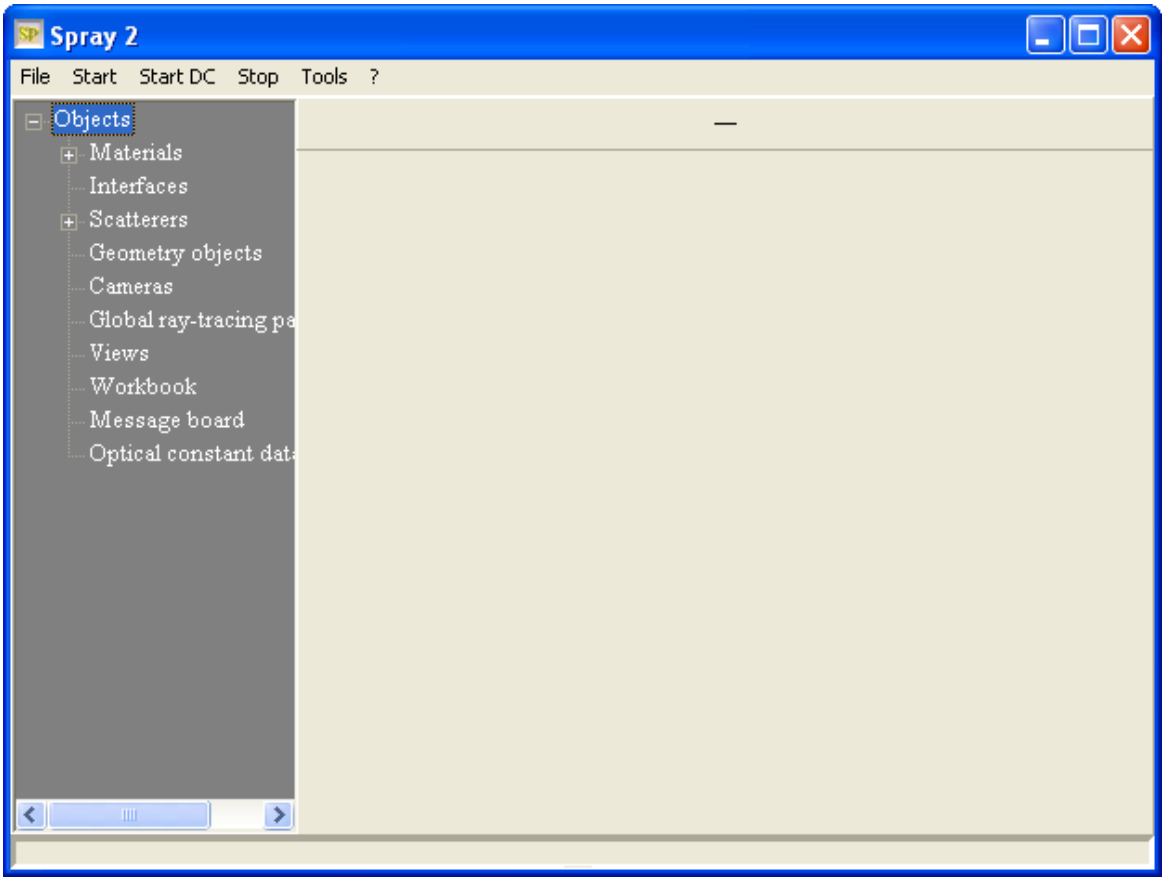
- First you will practise some basic skills with a simple point light source and a screen
- Then you will learn how to work with circular light sources and rendered views
- The important concept of 'interfaces' is demonstrated
- A cylinder is defined and covered with an interface, and a detector records the spectrum of the transmitted radiation
- The cylinder is coated with an absorbing material, the detector records an ATR spectrum (attenuated total reflection)

2.2 Point light source and screen

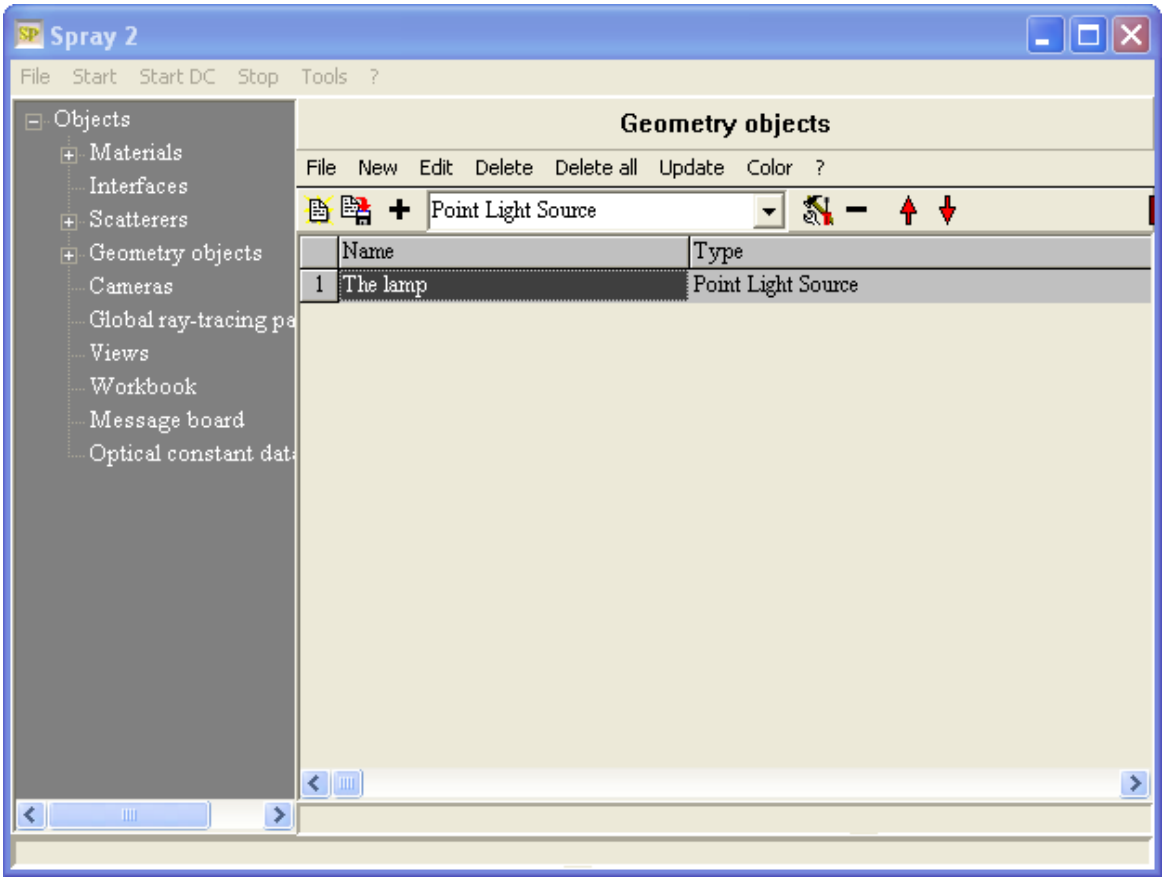
First we will do some introductory exercises which do not require the definition of optical constants. The goal is to inspect the output of a point light source with a screen. This example is simple enough so that we can guess how the result should look like. If we trust the model we can extend and modify it. Finally we will be able to do advanced simulations which cannot be done easily otherwise.
Start SPRAY:



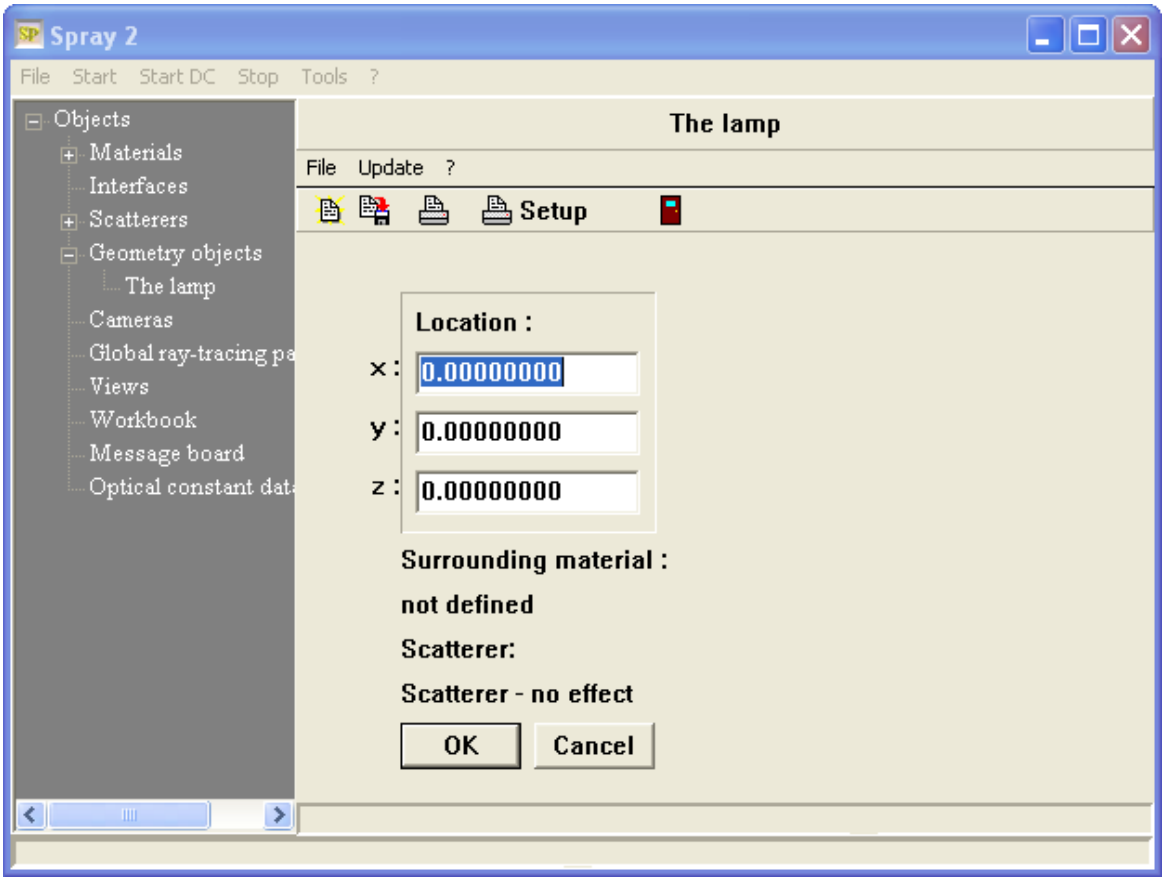
and press immediately F7 to enter the treeview level:



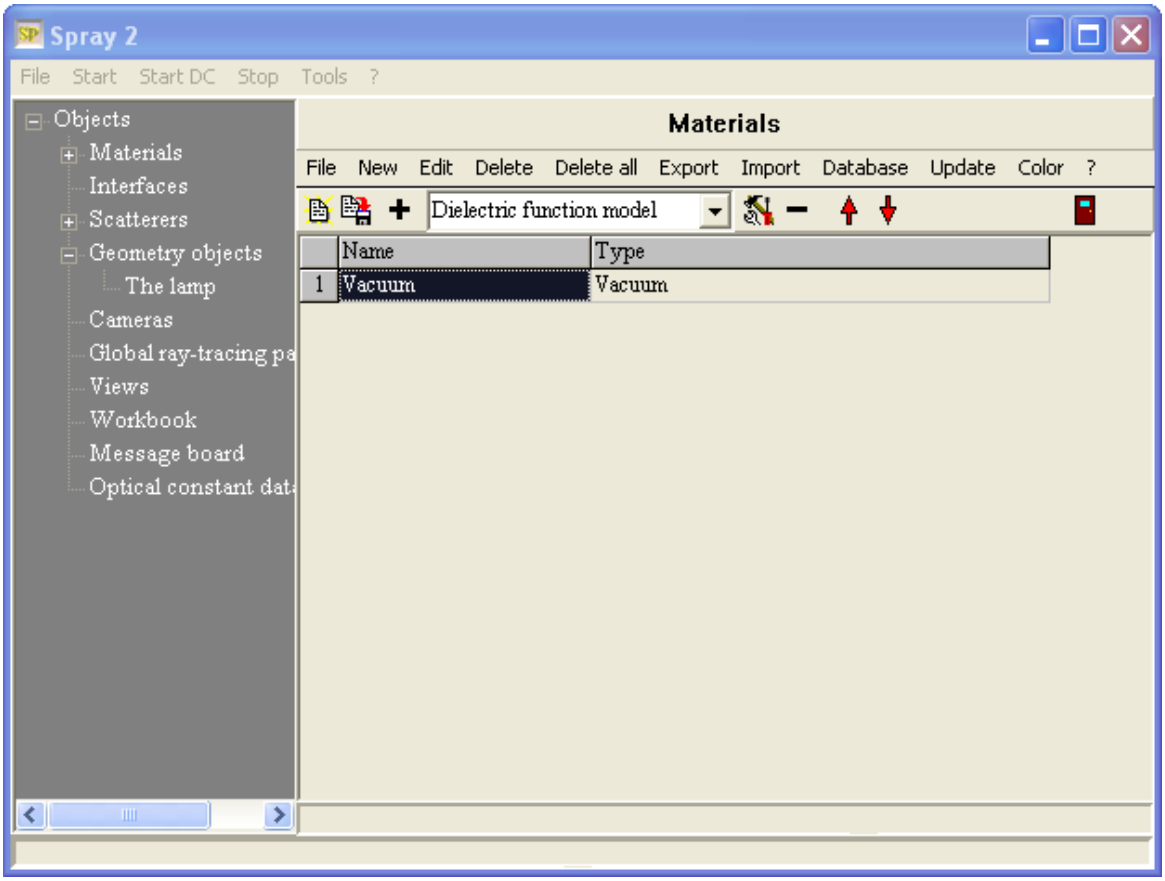
Right-click the branch called 'Geometric objects'. This command opens the object list in the right part of the main window. You can get general information on SPRAY lists in the technical notes section. The geometric object list is used to define the geometric objects in the setup. In between the '+' and the '-' button there is a dropdown list which is empty at the program start. Here you have to select the object type that you want to create. Select the entry 'Point Light Source'. Note that there is no object created yet. To do this you have to press the '+' button. A new object is created with a default name ('Point Light Source'). Click on the name and change it to a name of your choice, e.g. 'The lamp'. Press **Enter**, and the list now looks like this:



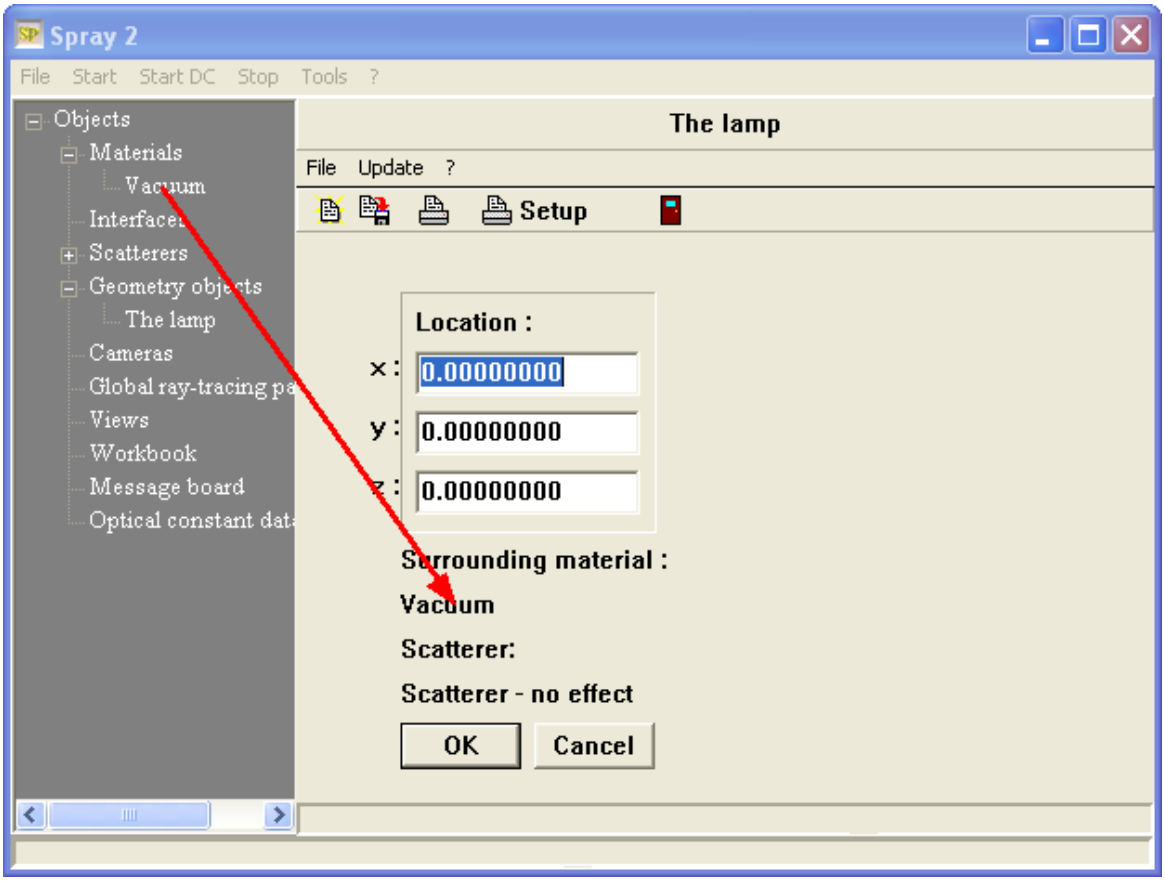
Note that the treeview branch 'Geometry objects' has a + sign to its left. You can expand this branch by a left mouse click on the +.
SPRAY setups have to have exactly one light source. The light source must be embedded in a certain material to have a well-defined starting environment for the rays. You have to set the light source material in a subwindow which also gives access to the geometric parameters of the object. You can open this subwindow by a right-click on the object's treeview branch:



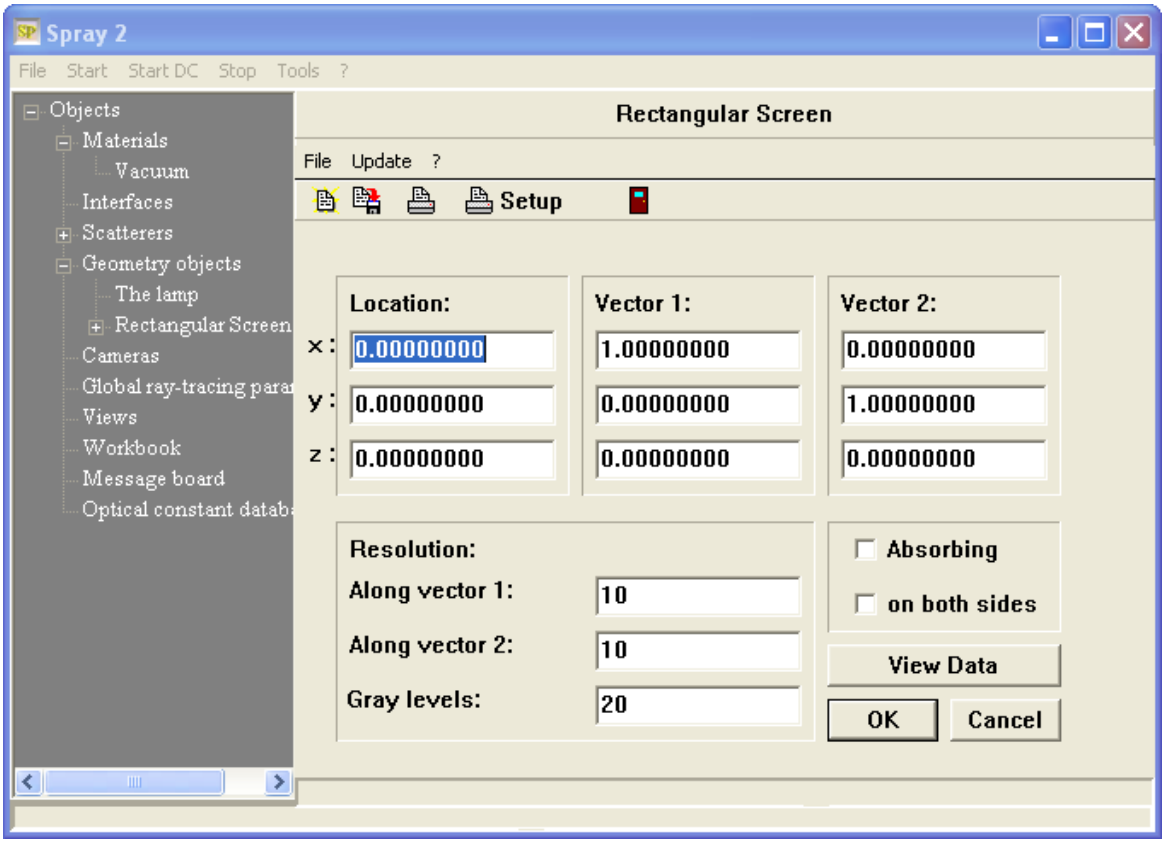
As you see there is not too much to do here. You can specify the x,y and z coordinates to define the position of the light source. We can leave the lamp at the origin for the moment, but we have to define a surrounding material. You have to do this by dragging a material from the list of dielectric functions (optical constants) to the text 'not defined'.
To get access to the list of materials right-click the Materials treebranch. The following list opens:



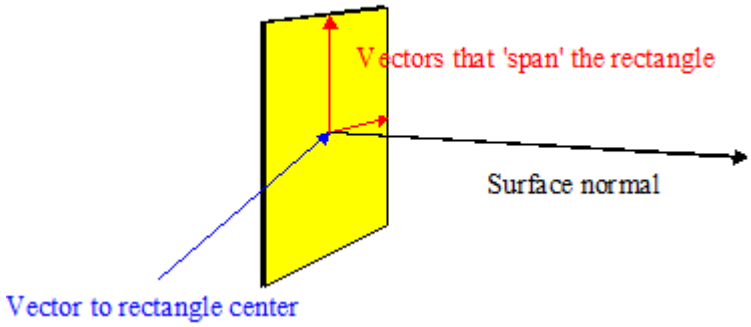
There is already the entry 'Vacuum' which we can use. To assign the material 'vacuum' to the point light source, proceed like this: Expand the treeview branch Materials so that the 'Vacuum' object appears in the treeview. Then right-click the treeview object 'The lamp'. Finally drag the 'Vacuum' object in the treeview to the text 'not defined' under 'Surrounding material'. Drop it there and verify the assignment:



Now we need an object to check if the light source works as expected. We will use a screen object to do this. Screens are very important to check a setup, so you should get used to apply screens right from the start. Go to the list of geometric objects (right-click the treeview item **'Geometric objects'**) and set the drop down list (Current entry is still 'Point Light Source') between the '+' and '-' buttons to 'Screen'. Then press the '+' button to create a new screen object. Open the new entry named 'Rectangular screen' by a right-click in the treeview:

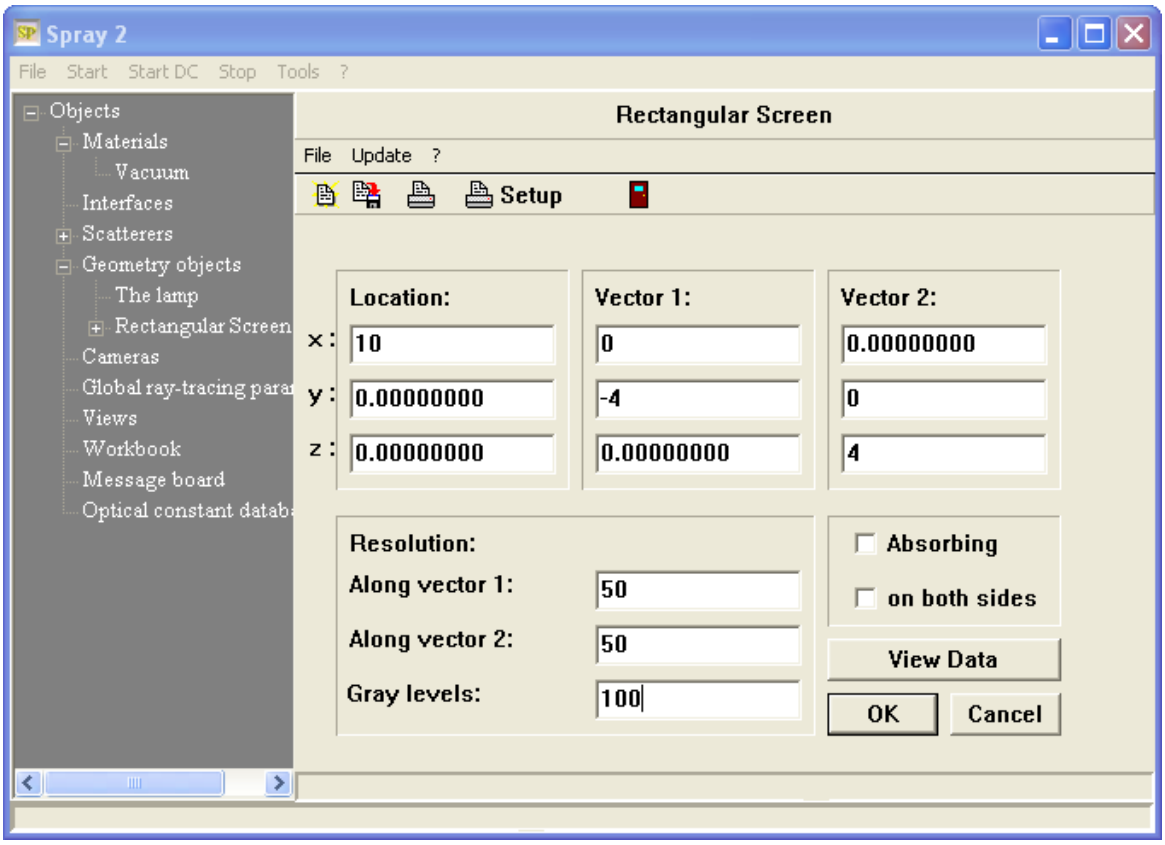


Compared to the point light source we have to set a lot more quantities to define the screen. The geometric position is defined by the center of the screen rectangle given by the x,y and z values of the 'Location' column (The blue vector in the picture below). The two other vectors (labeled 1st and 2nd) define the orientation and the size of the screen rectangle according to the following sketch:

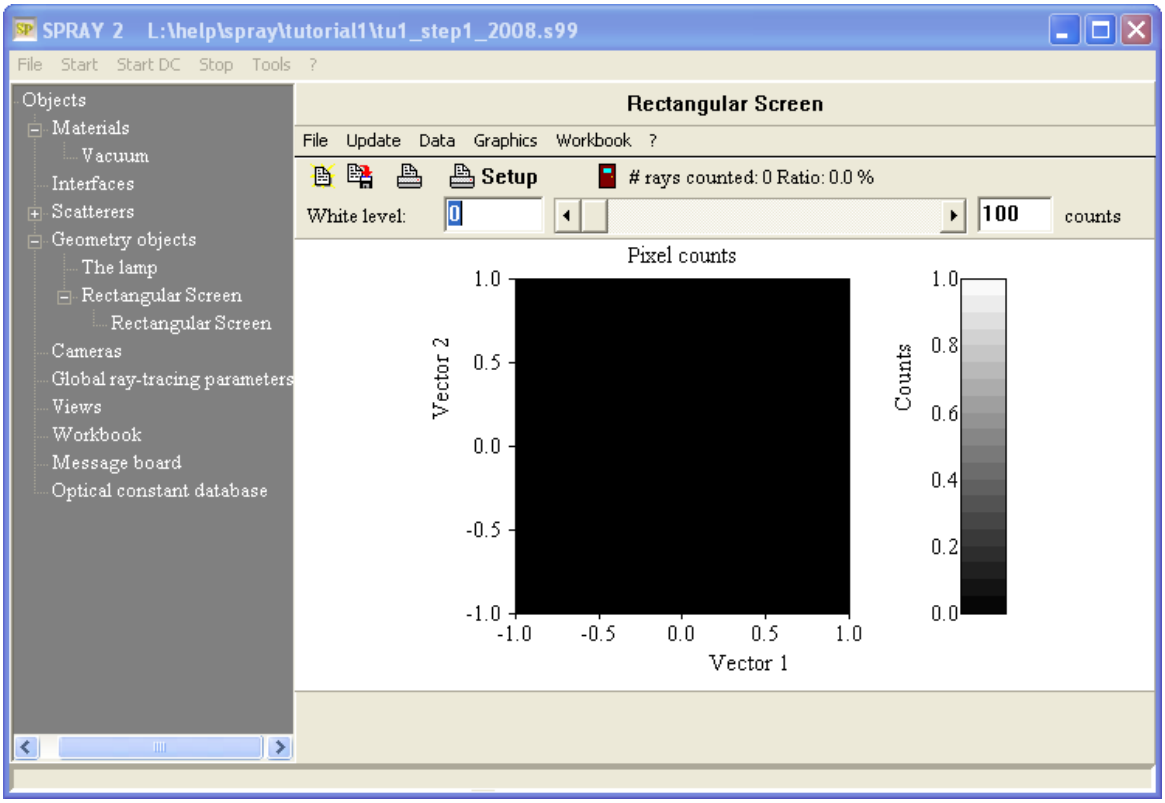


The two vectors should be perpendicular to each other, their cross product defines the surface normal.
The parameters of the 'Resolution of Data' section define the number of pixels in the direction of the 1st vector (labeled as 'x') and 2nd vector ('y'), respectively. The 'Color' parameter determines the number of color levels for the graphical output. You will see immediately what this means.
During a simulation, screens count the number of rays that hit their individual pixels. The setting of the 'Absorbing' checkbox determines if the screen absorbs rays or not. If the screen is non-

absorbing the rays are detected by the screen but continue their way as if the screen were not there. If 'on both sides' is checked the pixels count rays independent of their direction. Otherwise, only those rays that arrive from the surface normal side are detected.
To define a screen in the y-z-direction (dimensions: 8 cm by 8 cm) moved away from the origin by 10 cm in the x-direction you have to set *Location* to (10,0,0), 1st vector to (0,-4,0) and 2nd vector to (0,0,4). This way the surface normal points in the negative x-direction. Set the number of pixels in both directions of the screen to 50, the number of gray levels to 100:



Press **OK** when you are ready with your settings.
Expand the object 'Rectangular screen' in the treeview and right-click its subobject (which is also called 'Rectangular screen'):

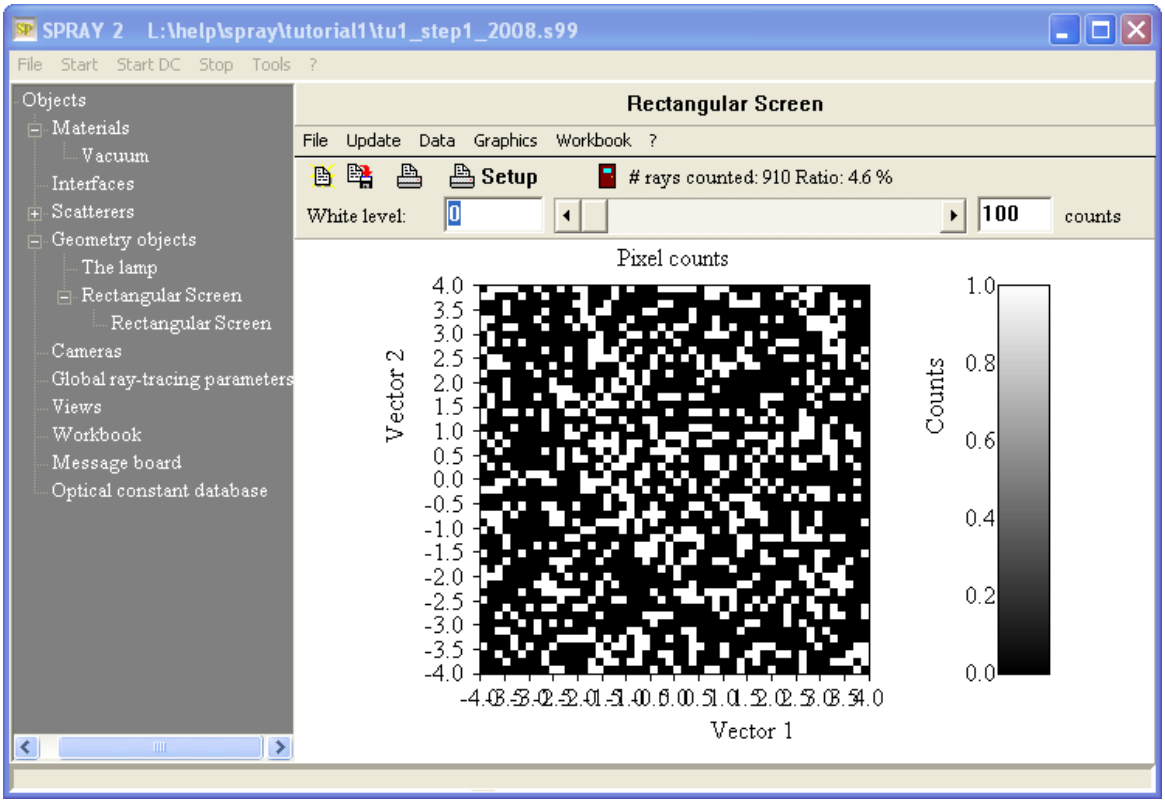


You see graphical representation of the screen. Since no simulation has been done yet, the screen is black up to now:

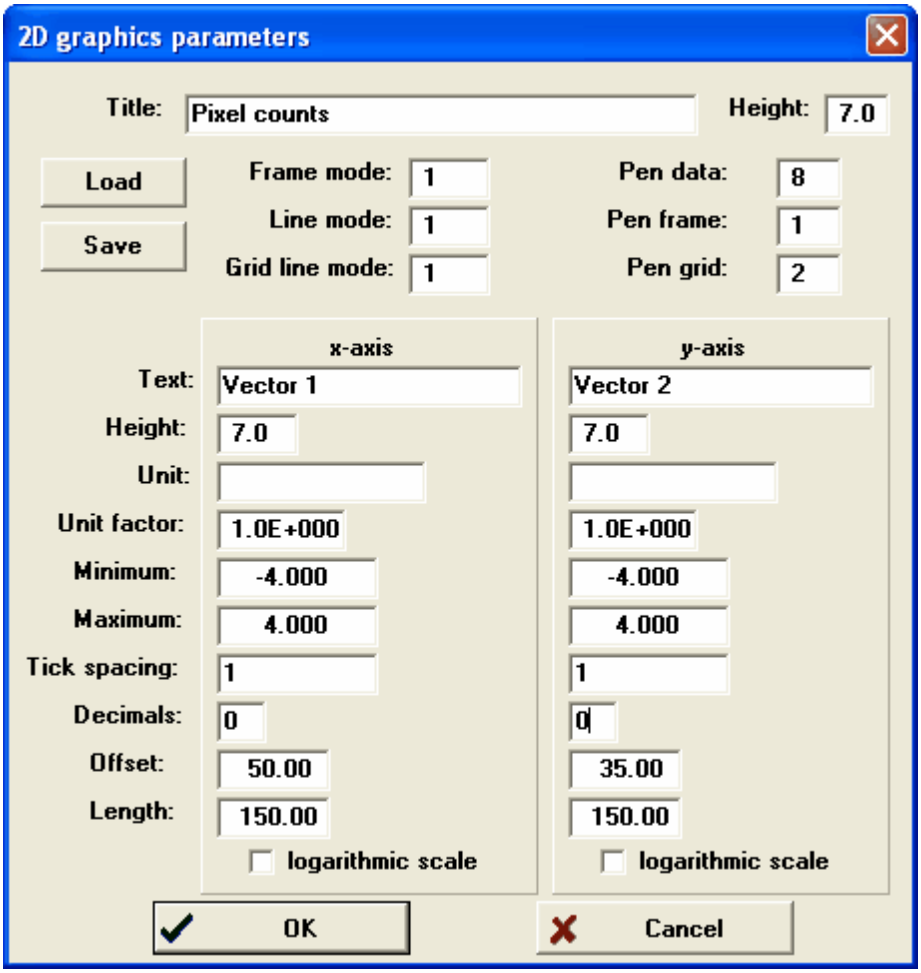
We are almost ready for our first simulation. However, before we start the computations right-click the treebranch **'Global ray-tracing parameters'**. This opens a subwindow for setting some important quantities.
In the '**Spectral range**' section you can set the spectral range for the ray tracing simulation. Specify the minimun, the maximum, the number of spectral points and the unit. For the latter you can choose between 1/cm (wavenumbers), nm (wavelength), eV (energy), micron (wavelength) and THz (frequency).
The '**Angle resolution**' is important for interface objects. These compute (before the simulation is started) the angular dependence of the reflectance and transmittance using the specified number of points for the range 0 ... 180 degrees. Usually this does not take too long. Hence there is no reason to go to a low angle resolution unless you have a large number of interfaces in your model.
The '**Number of photons per spectral point**' determines how many rays are processed for each spectral point. How many rays you need depends very strongly on the questions that you have to answer.
Finally you can set the '**Max. number of interactions**' for each ray. After a ray has been emitted by the light source SPRAY counts how many interactions this ray has with the objects of the setup. If the specified maximum value of interactions is reached before the ray reaches infinity or is absorbed the tracing of this ray is stopped. This is to avoid situations where a ray is reflected back and forth forever between two ideal mirrors. Usually you do not have to change this value unless you study very special setups.
Set a spectral range of 500 ... 5000 1/cm (we will work in the infrared in the next steps) with 20 spectral points, 180 points angular resolution and 1000 photons per spectral point.

Before we start the simulation based on the present model you should save the complete SPRAY configuration with **File|Save As** in the main window. The present configuration can be found in the file tu1_step1.s99. If you have problems reproducing the next results please load this demo configuration and compare it to yours.

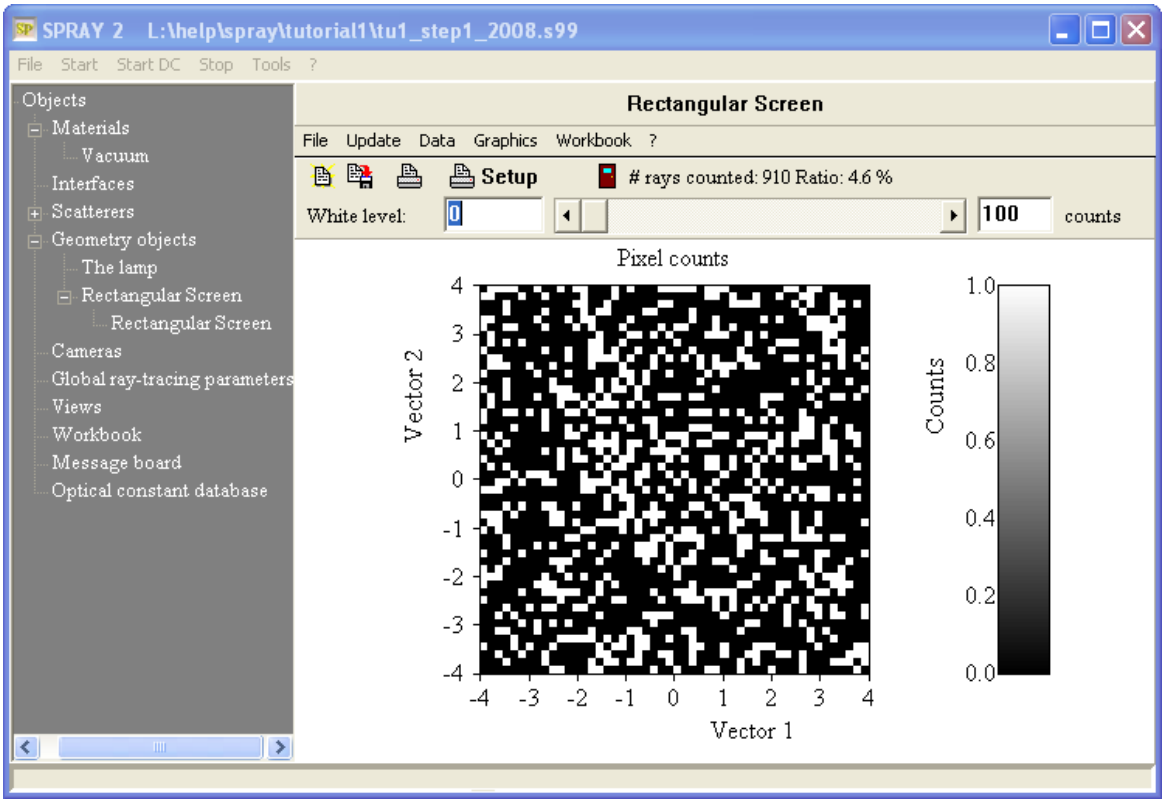
Now use the **Start** menu command to start the ray-tracing run. A progress bar tells you the present status of the computation. If it is finished you can look at the screen window to see how much radiation arrived. If this window is not on the screen any more you have to right-click it in the treeview:



If you like you can use the **Graphics|Edit graphics parameters** dialog to change the settings of the axis labels. Here is an example:

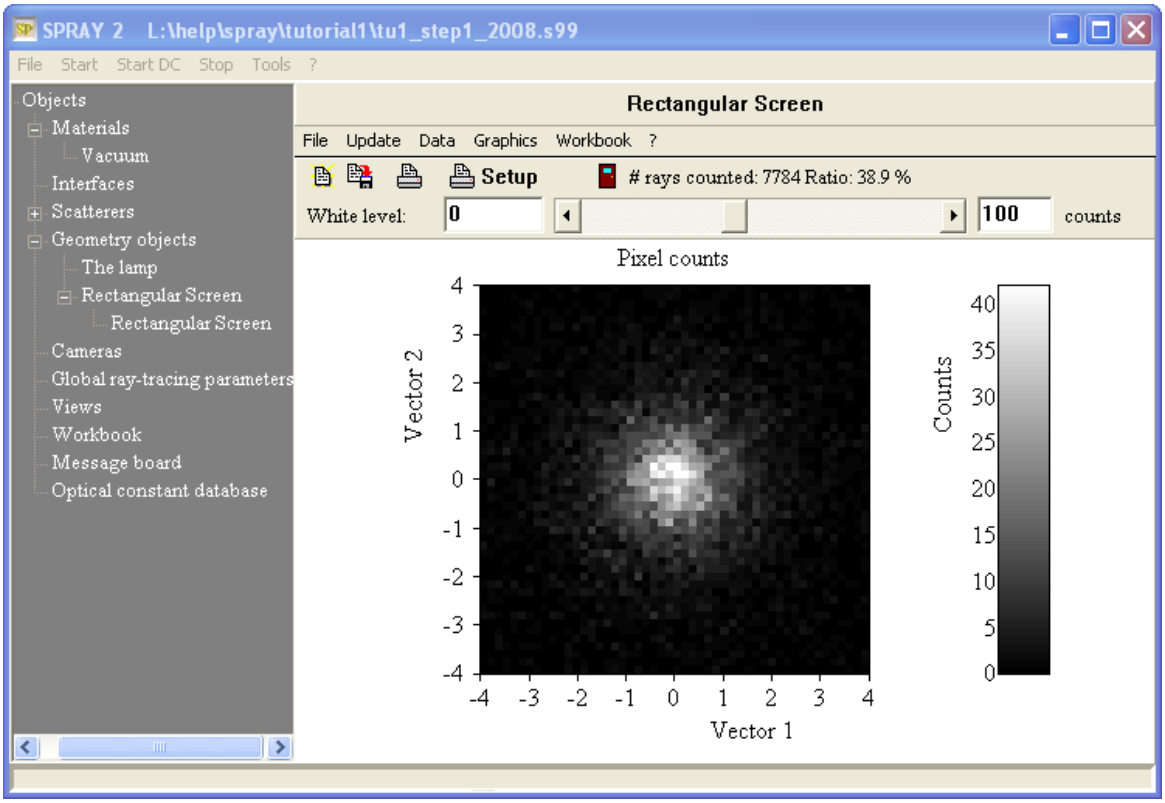


The graph now looks a little nicer:



Screens are not wavenumber selective, they count every ray hit independent of the frequency. From the 20000 emitted rays 910 have reached the screen (4.6%). The distribution is quite uniform.
You can modify the scaling of the grayscale plot by moving the slider on top. The number to the left of 'counts' gives the number of photons per pixels which are plotted as 100% white.

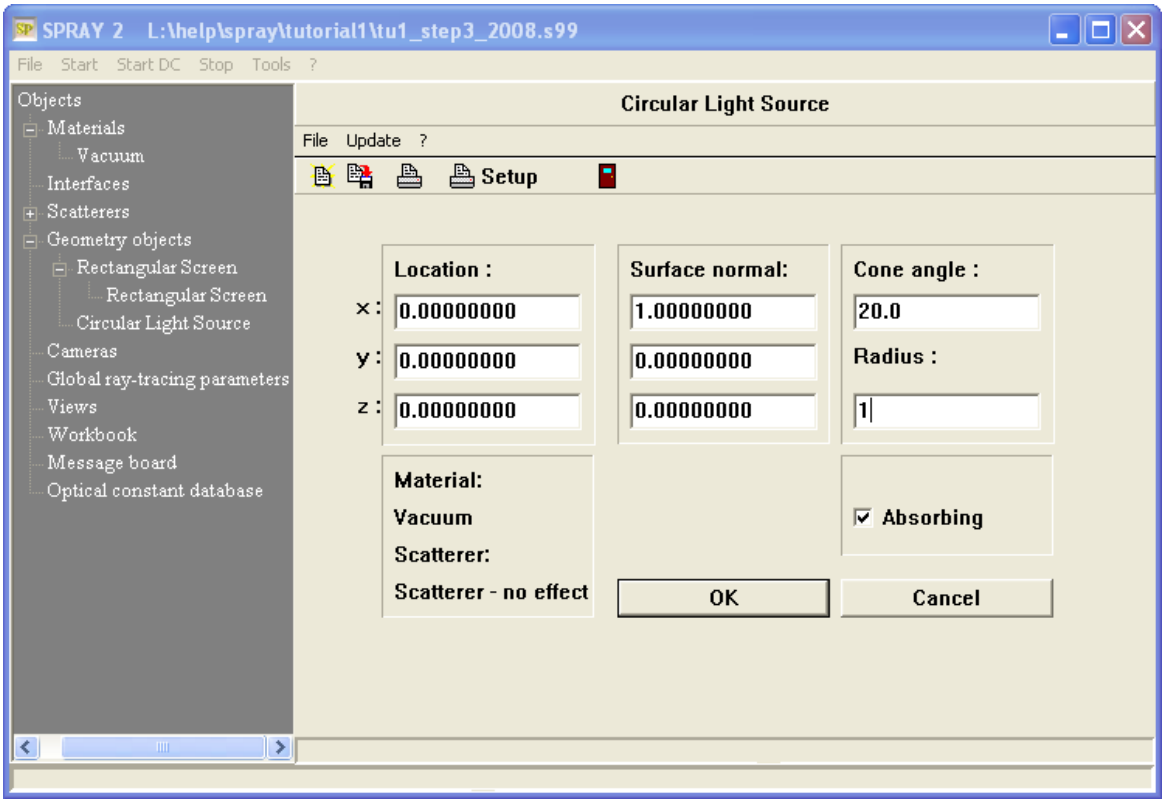
Now we place the screen a little closer to the light source. The radiation distribution should be less uniform, with a higher concenctration in the center of the screen. More rays should hit the screen since it will cover a larger solid angle now.
Change the x-coordinate of the screen location from 10 cm to 1 cm (do not forget to press the **OK** button) and then press **Start** again. Now the screen looks like this (20 rays/pixel as 100% white level):



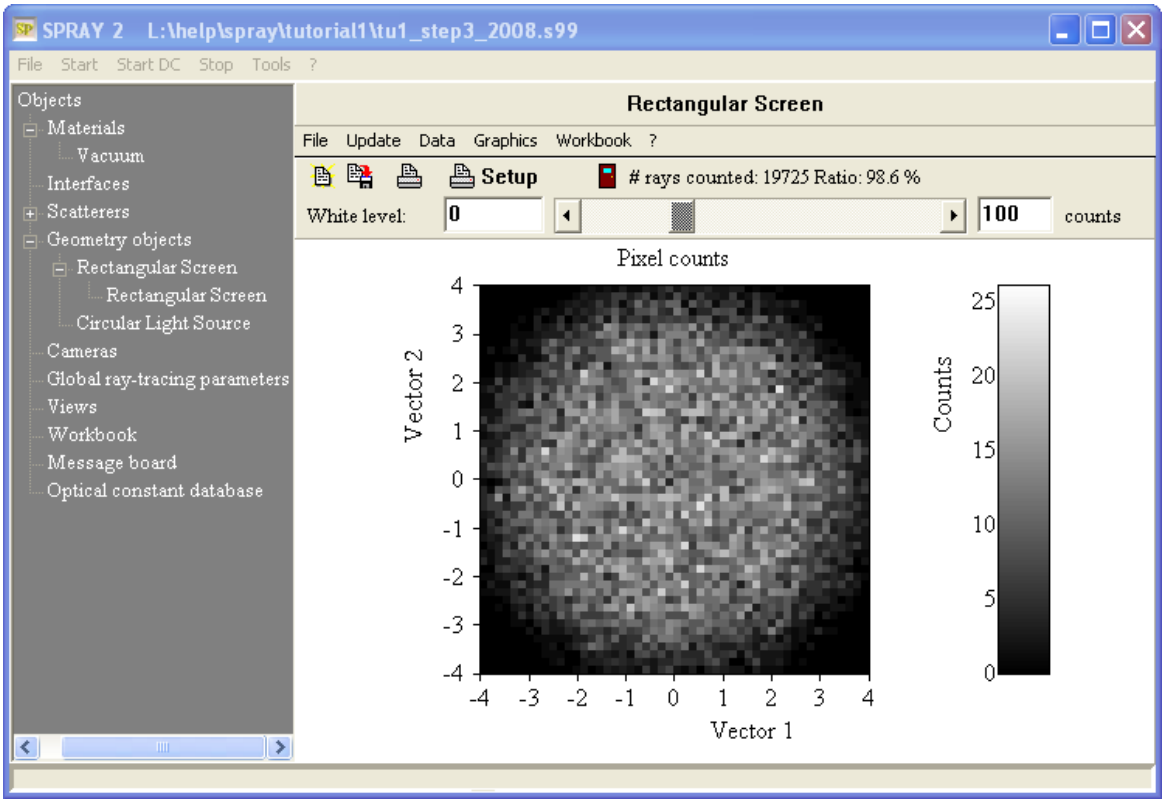
Note the larger fraction of rays reaching the screen (38.9%). This configuration was saved under the name tu1_step2.s99. If you cannot reproduce the above result, please load this configuration and compare it to yours.
Finally we can check if the screen gets 50% of the rays if it is placed very close to the light source. Since the light source emits isotropically half of the rays should hit the screen in this case. Set the x-coordinate of the screen's position to 0.001 cm and re-run the simulation. I found a value of 49.9% (tu1_step3.s99).

2.3 Circular light sources and rendered views

In many cases it is advantegeous to work with more or less collimated beams instead of point light sources. This is in particular important if you want to simulate only a part of a setup, e.g. a sample holder in a spectrometer. You could then start the simulation at an intermediate focus if you know the focus size and the beam divergence approximately.
In these cases you can use a light source of type *Circular light source*. Delete the point light source in the present configuration (by selecting its row in the object list and pressing the '-' button) and set the object type to 'Circular light source'. Create a new object of this type with the '+' button and open it (treeview right-click). Assign vacuum as surrounding material (see above how to do that by drag&drop) and set the radius to 1 cm. The dialog should now look like this:

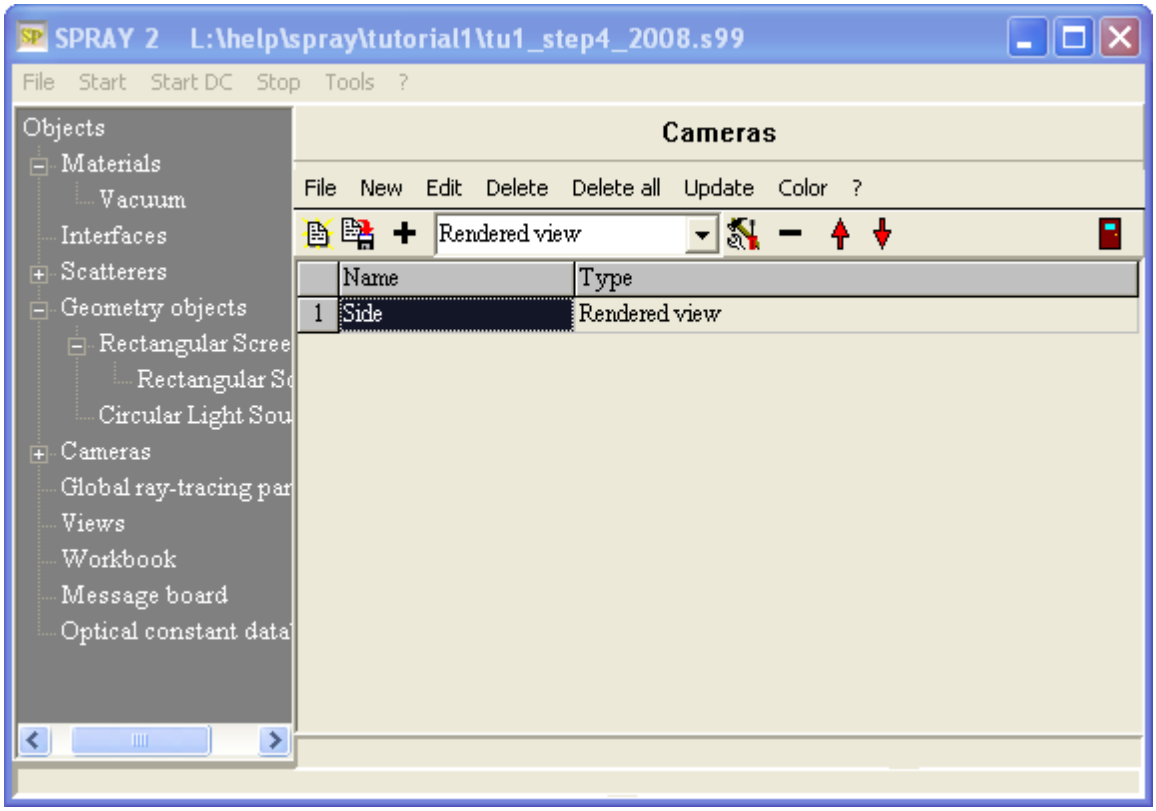


This defines a disk with radius 1 cm at the origin, emitting rays in a cone of 20 degrees opening angle around the direction given *by Surface normal* (i.e. the positive x-axis).
Now move the screen to its original position at x = 10 cm and re-run the simulation. The screen looks like this now:

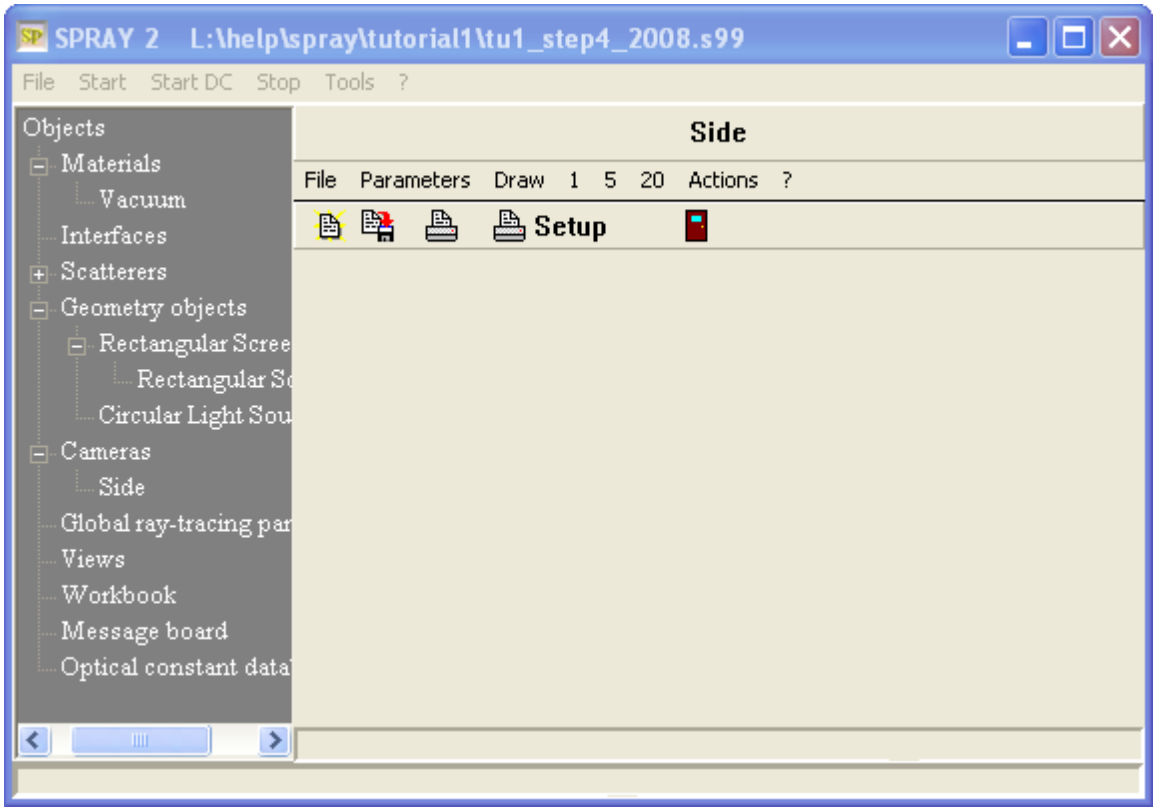


Almost all rays hit the screen now. This configuration can be found in the file tu1_step4.s99.

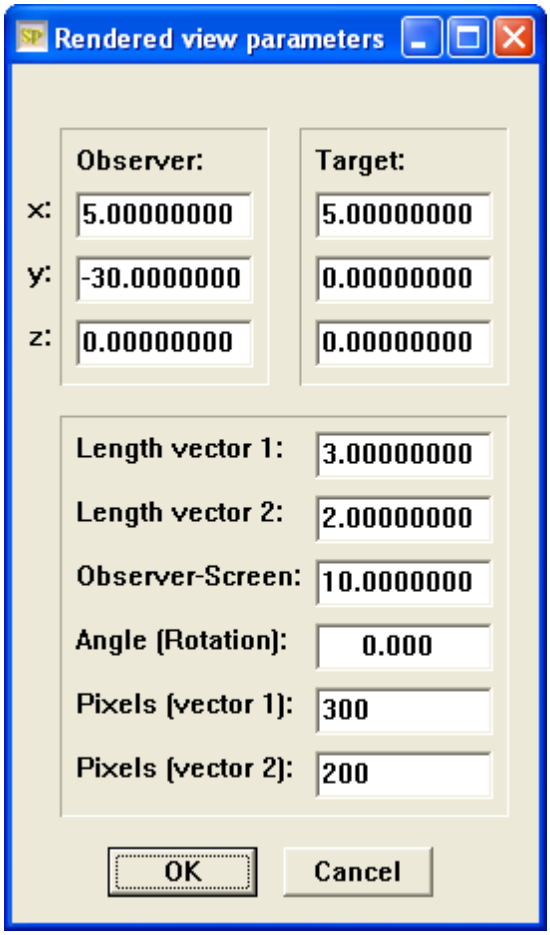
In addition to typing in geometry data to position your optical components in space you certainly would like to have a view on your setup. This can be done most conveniently using so-called camera views. These are 'snapshots' of your scenery which can be exported as bitmaps for documentation purposes. Here is a short introduction showing how to create and handle camera views.
Right-click the treeview branch **Cameras**. This list collects various views of your scenery. Select the type 'Rendered view' in the type selection dropdown list and create a new view object pressing the '+' button. Change the name of the new object to 'Side'. The camera list should now look like this:



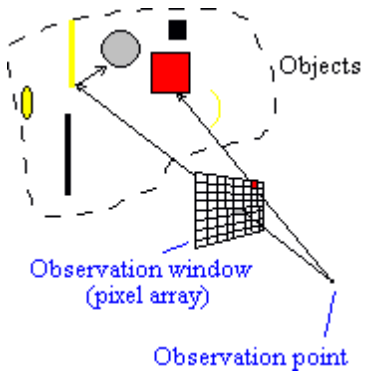
Open the new object in the treeview:



This window will show the view once it is computed. First we have to set a lot of parameters in the dialog that opens after the **Parameters** menu command:



The 'photograph' of the scenery is taken by an 'Observer' whose position is given by the vector called **Observer**. The observer looks towards a target point the coordinates of which are specified in the **Target** column.
Like in the following sketch there is a virtual pixel array between the observer and the scenery. The distance between the observeration point and the center of the screen is set by the '**Observer-Screen**' parameter:

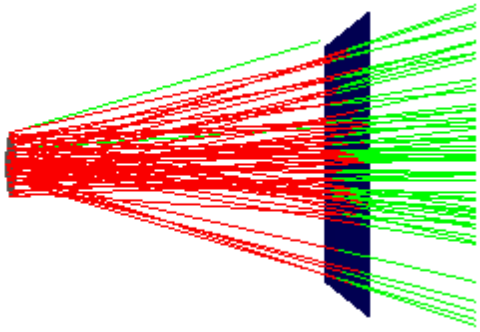


The horizontal extension of the screen is given by **Length vector 1**, the vertical one by **Length vector 2**. The screen is adjusted by 'gravity', i.e. the vector 1 direction is horizontal (perpendicular to the z-direction). However, if you like, you can rotate the screen by specifying an angle different from zero (parameter **Angle**).
Finally you have to set the resolution of the screen by specifying the number of pixels in the two

screen directions. The more you take here, the more details you will see, but the longer you will have to wait for the computation of the picture.

If you set all parameters like in the dialog example given above you will see a side view of the scenery. After the **Draw** command the picture is computed and displayed in the rendered view window:

To the left you see the circular light source, to the right the screen.
In a rendered view, you can also have some action in the sense that you can emit some test rays from the light source and watch where they go. The '**1**' command sends 1 ray, the '**5**' command 5, and the '**20**' commmand sends 20, of course. You can execute any of these commands several times to accumulate as many rays as you like:

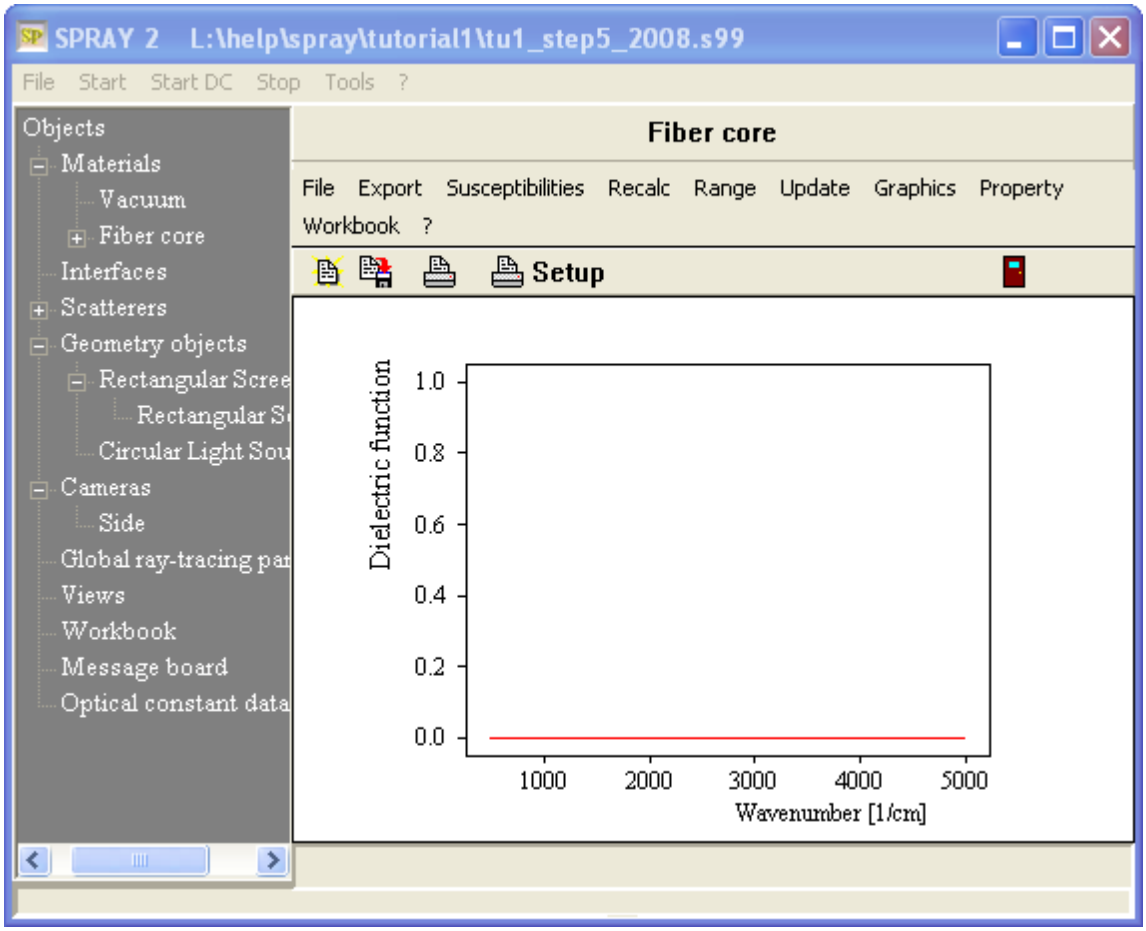


The rays that hit some of the objects are drawn in red, those that escape to infinity are drawn green.
If you want to clean a rendered view from the test rays you have to activate the **Draw** command ones more.
If you have specified more than one rendered view the test rays are drawn in all views simultaneously.

This configuration can be found in the file tu1_step5.s99.

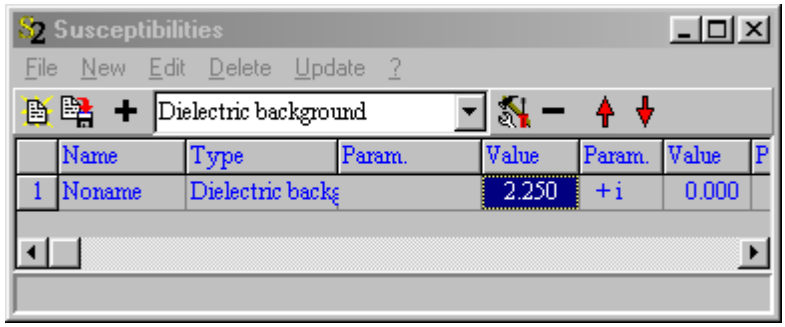
2.4 Defining interfaces

Optical rays change their direction if they hit an interface between materials of different refractive index. Hence interfaces play a very important role in ray-tracing simulations. In the following we will simulate how radiation travels through a solid cylinder covered by a thin absorbing film. However, before we define our first SPRAY interface we need to have at least two different materials (or refractive indices). Up to now our list of materials contains vacuum only. Hence we have to create a new entry. Setting up new materials is not easy since you have to know a lot about materials and their optical constants. For the present exercise, however, a very simple model will do the job. For the cylinder material we will use a constant and real dielectric function, which means that there is no dispersion and no absorption. In this case the refractive index (which is the complex square root of the dielectric function with positive imaginary part) is also real.
Open the list of materials. Select the object type 'Dielectric function model' and press the '+' button to create a new dielectric function object. Overwrite 'dummy name' by 'Fiber core' and open the new object in the treeview. Select the **Range** command and set a spectral range 500 ... 5000 1/cm with 100 data points. Then press the 'a' key on the keyboard for an automatic scaling of the graph. You should have arrived here:

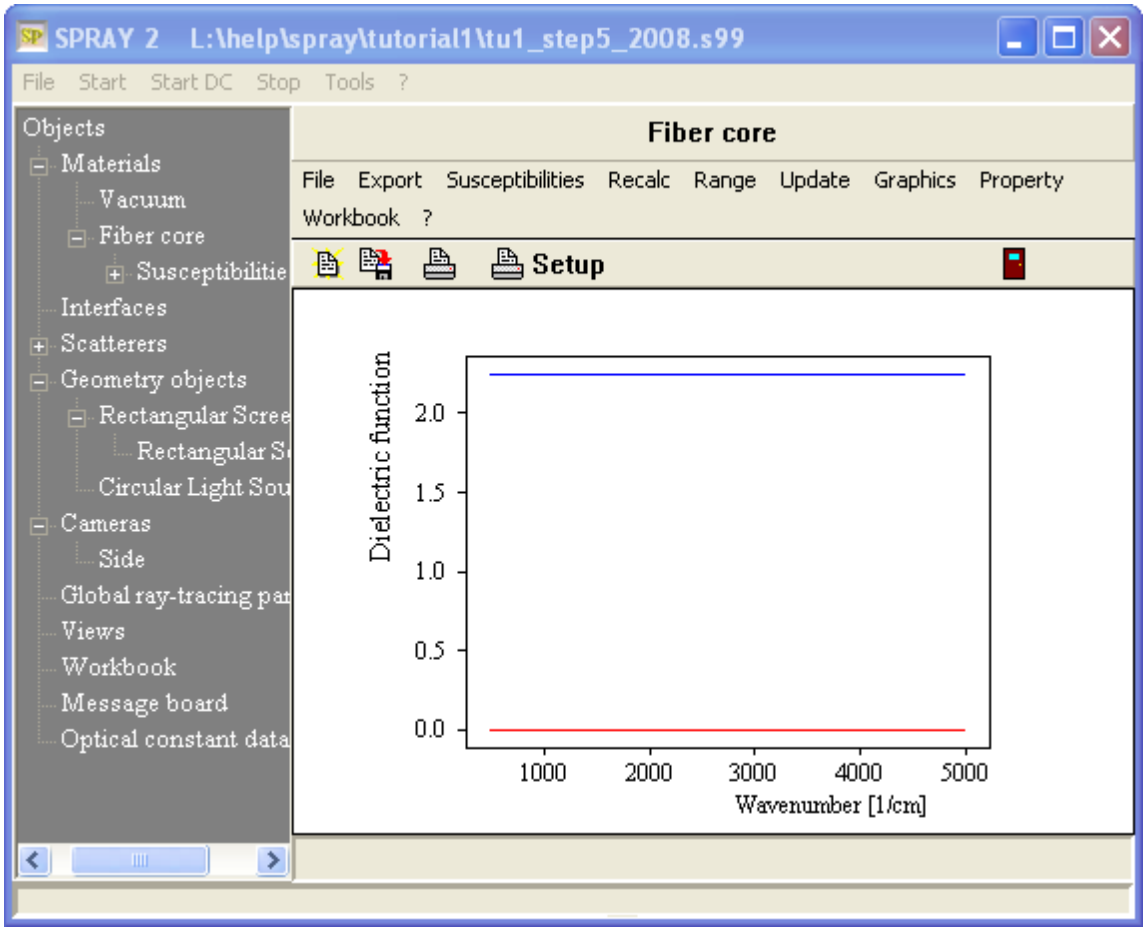


The new object has a subbranch in the treeview called **Susceptibilities** to open another list which contains the susceptibility contributions to the dielectric function. Select there the new type 'Dielectric background' and create an entry of this type by pressing '+'. Move the cursor to the

'1.000' cell and change it to '2.25':



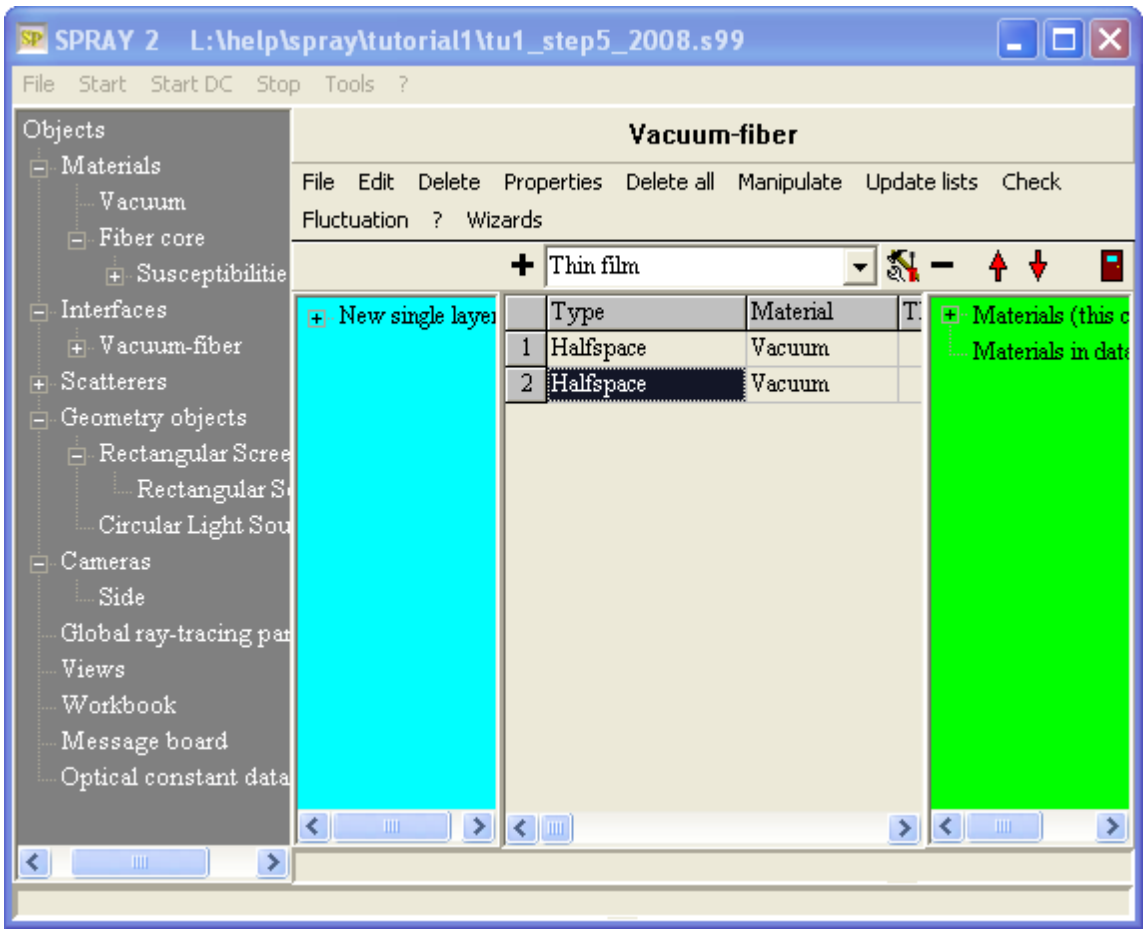
Close this list and select 'Recalc' in the 'Fiber core' window, followed by another 'a' for automatic re-scaling:



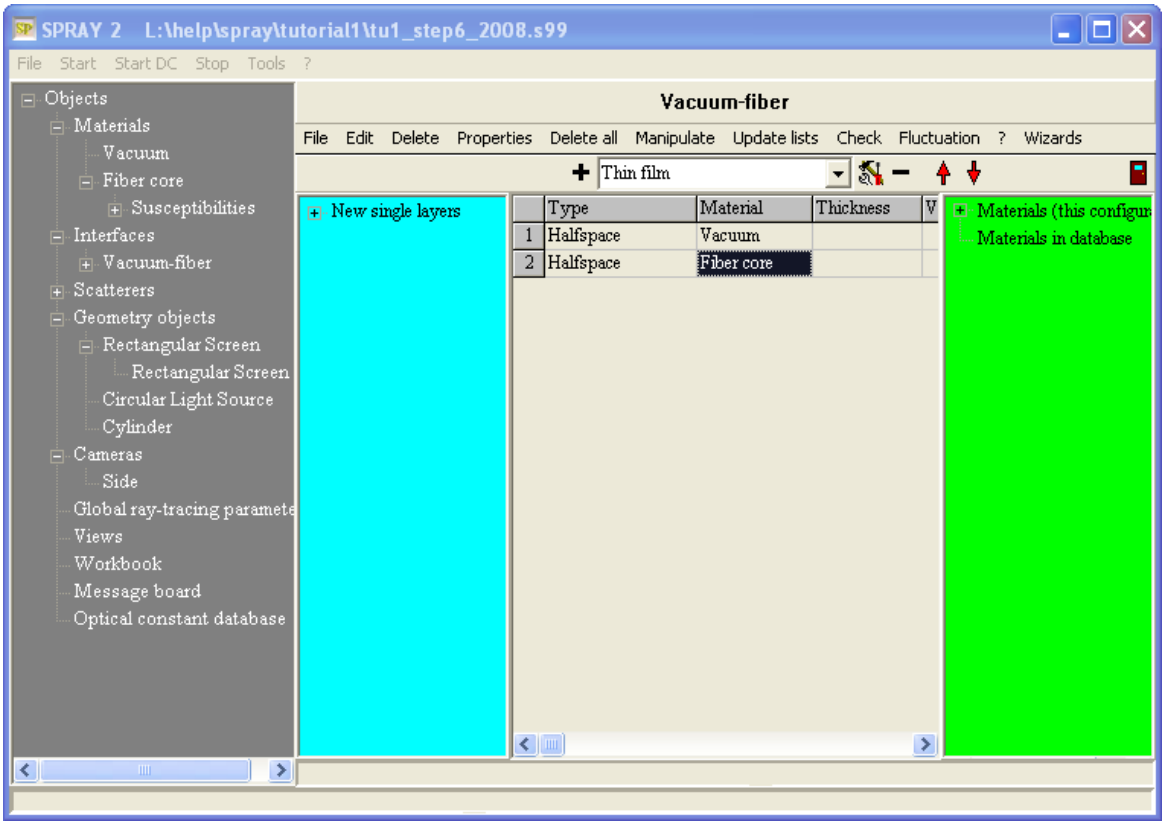
The blue line shows the real part of the dielectric function (=2.25) whereas the red line displays the imaginary part (=0). The corresponding refractive index of this fiber core material is 1.5, like that of glass in the visible spectral range. Later on we will create more complex optical constants but for the moment this one will be good enough.

We can now define the first interface that we will use for the cylinder walls in the next section. Open the list of interfaces by right-clicking 'Interfaces' in the treeview.
Select the new object type 'Layer stack' and create an item of this type ('+' button). Rename the

new entry (initially called 'Dummy name') to 'Vacuum-fiber'.
Open the new layer stack object in the treeview. You are now here:



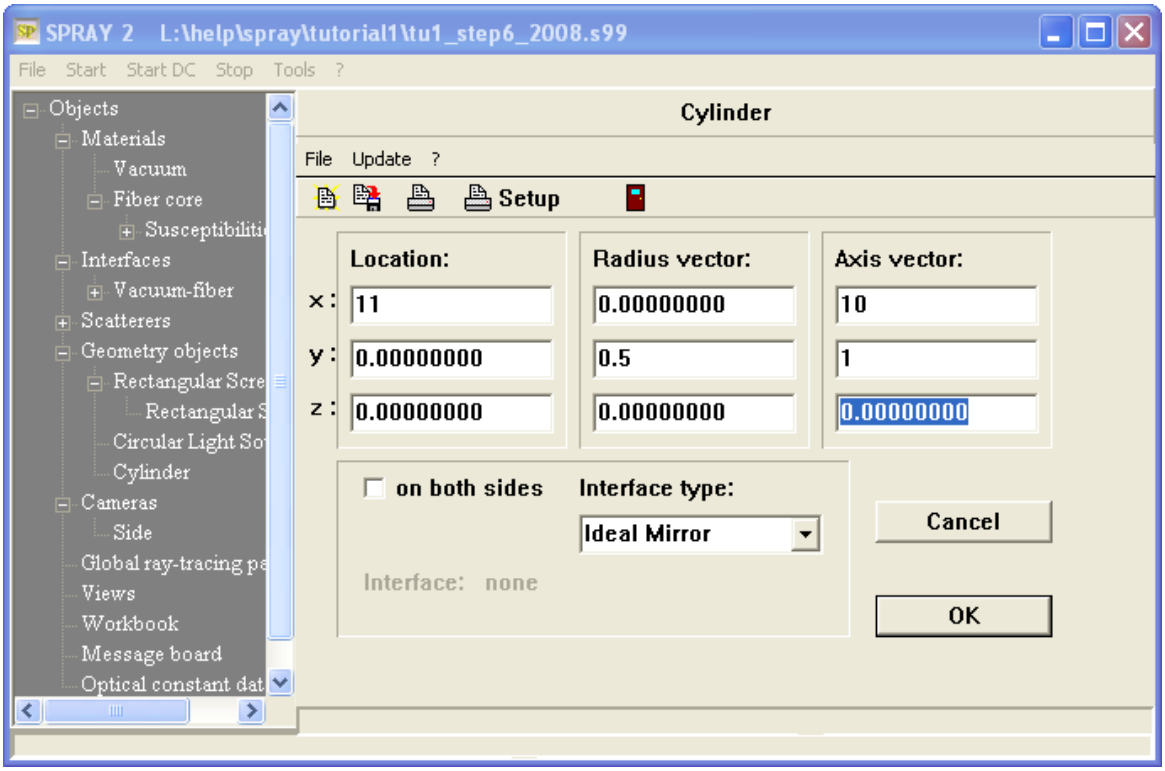
The layer stack defines an interface between the top halfspace and the bottom halfspace. Later we will add a thin film (representing a coating of the cylinder) but at present we just want to replace the vacuum entry of the bottom halfspace. Drag&drop it from the treeview:



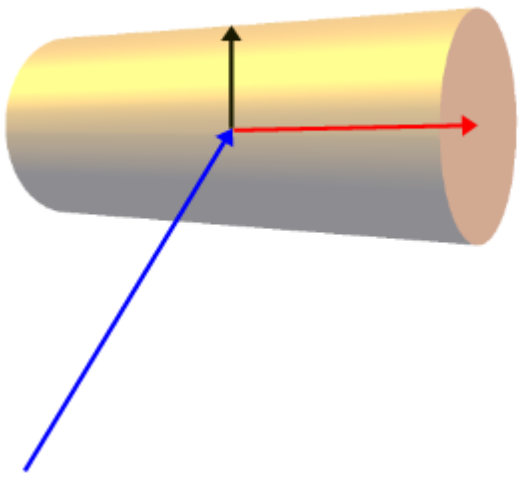
We are now ready to 'coat' a geometric object like a cylinder with the interface we just defined. (The present status can be found in the configuration file tu1_step6.s99).

2.5 Adding a cylinder and a rectangular detector

Now we will add a cylinder to the scenery which is placed between the light source and the screen. The cylinder is going to represent an optical fiber used to guide light from one side to the other without reflection losses. This is achieved by making use of the total reflection of light that occurs if the angle of incidence is above the critical angle.
Open the list of geometric objects and create a new object of type 'Cylinder'. Name it 'Fiber'. To set its geometric parameters open it in the treeview:

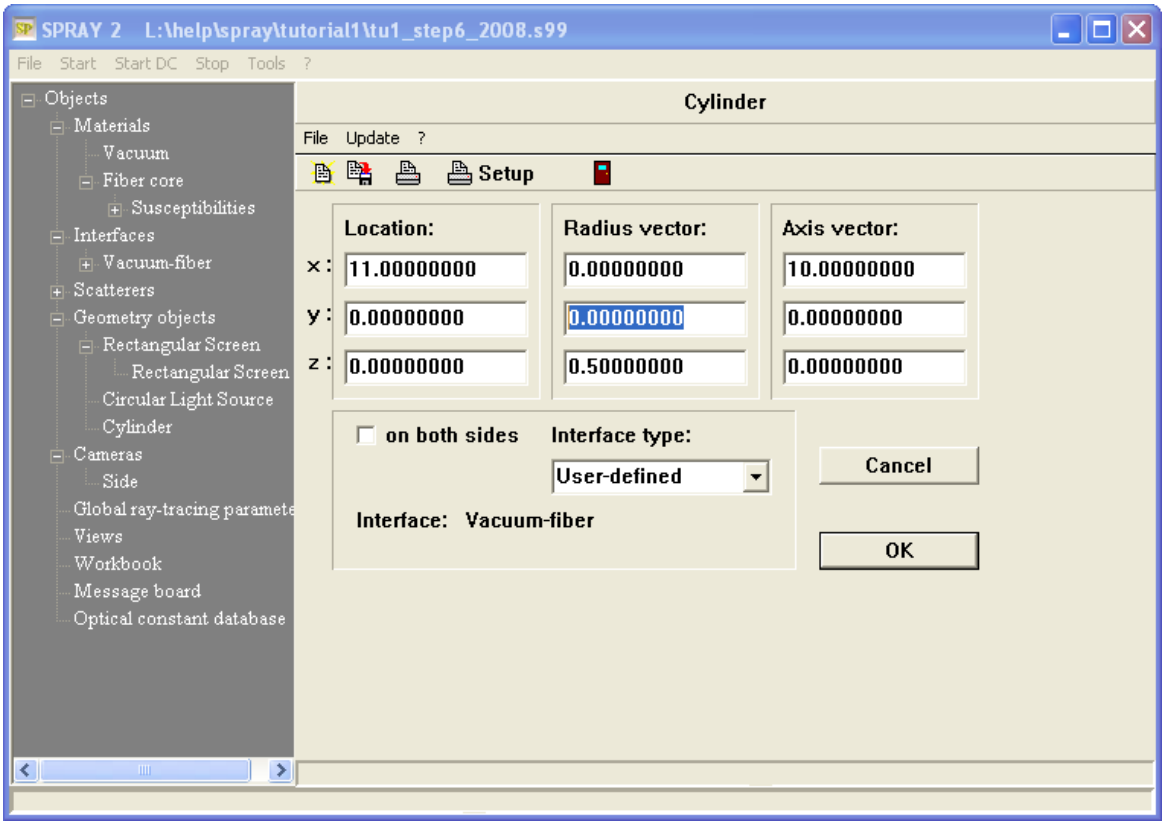


The vector called 'Location' points to the center of the cylinder (drawn in blue in the sketch below). The 'Radius vector' (black vector) points from the center to the cylinder surface. It is perpendicular to the cylinder axis. The 'Axis vector' (displayed in red) is parallel to the cylinder axis and points from the cylinder's center to the center of one of the circular end areas:

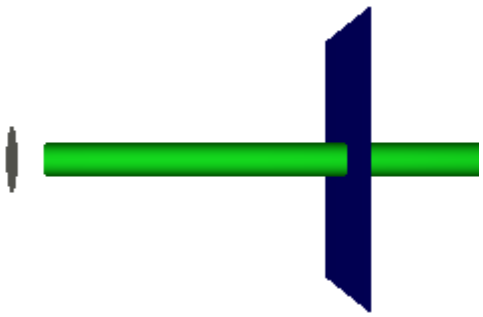


The white area to the right of 'Interface' shows the name of the interface that is assigned to the cylinder. The default setting 'none' must be changed, of course. You do that by a drag&drop operation, dragging an interface from the treeview to the white area. The dialog above shows already all settings that are to be done in our case. Try to reproduce all values, in particular to drag the 'Vacuum-fiber' interface to the desired area. You have now defined a 20 cm long cylinder with 1 cm diameter, covered with an interface that represents the transition from vacuum

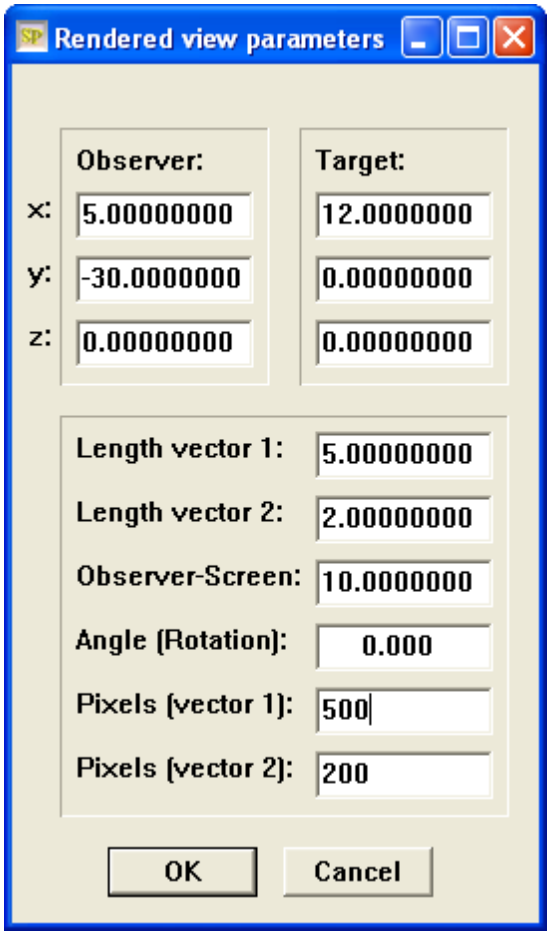
to the fiber core material:



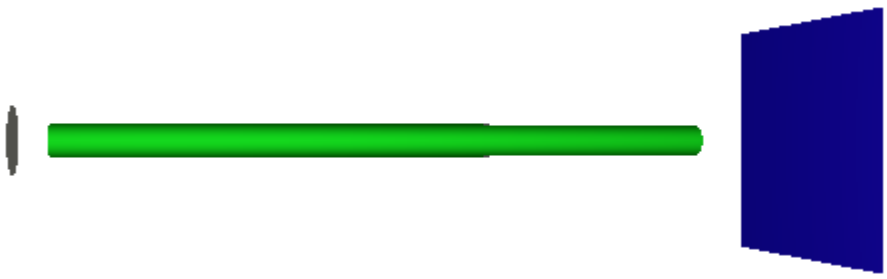
Looking at the rendered view called 'Side' we see the following situation:



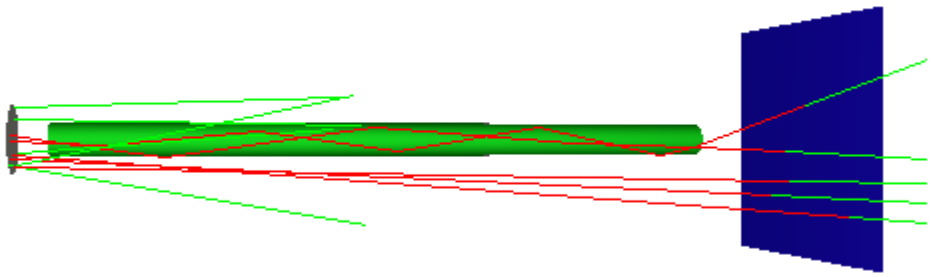
Move the screen to x = 25 cm and re-arrange the rendered view settings according to the following dialog:



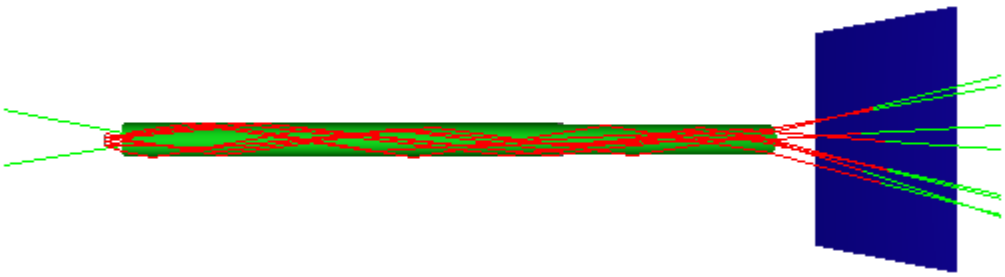
Now you can view the complete cylinder (after using the Draw command):



Try a few test rays and check (qualitatively) that the wave-guiding by total reflection works:

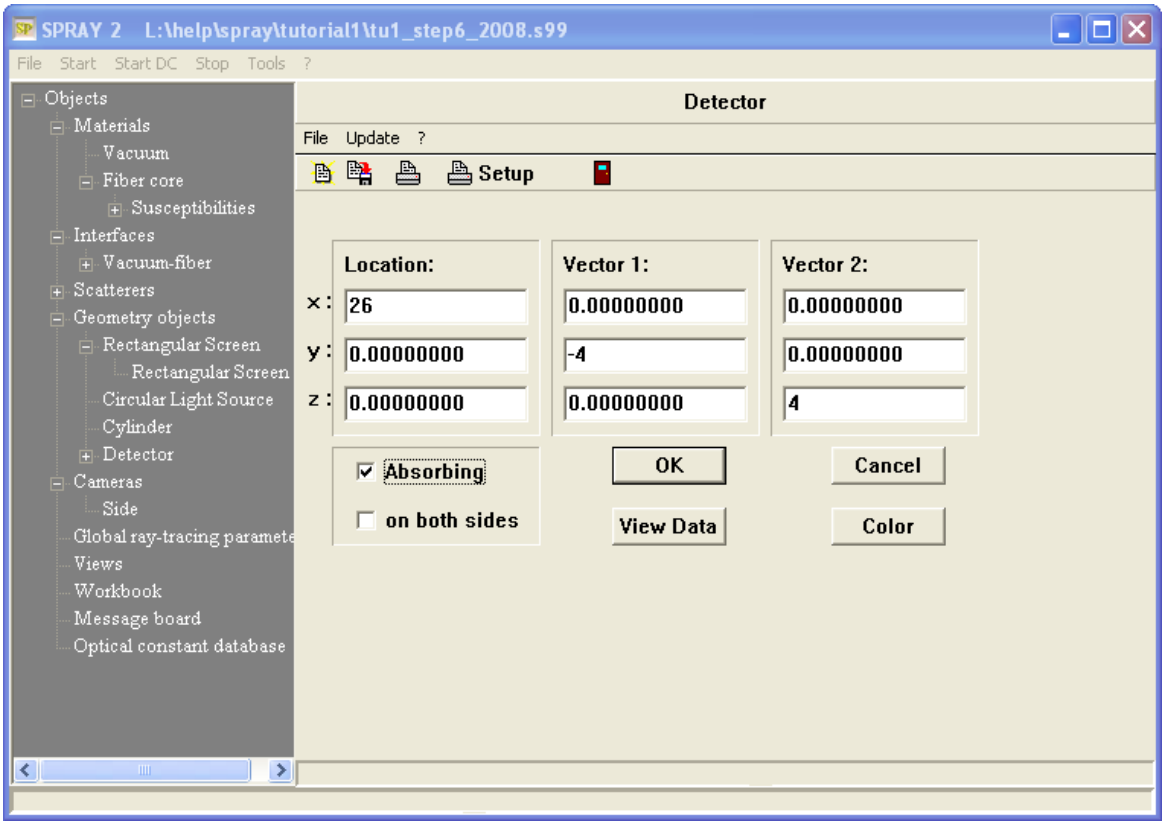


Now change the light source geometry: The location should be (0.5, 0, 0) (a little closer to the fiber head) and the radius is reduced to 0.2 cm. Now all emitted rays hit the fiber front end:

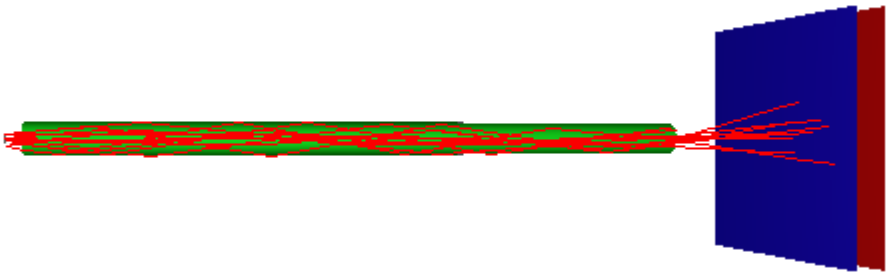


Go to the main window and press the 'Simulation' button to start a ray-tracing simulation. Look at the screen. It should have received roughly 92% of the radiation which is expected: The reflection losses entering the fiber are about 4%, and the same amount is lost at the other side.

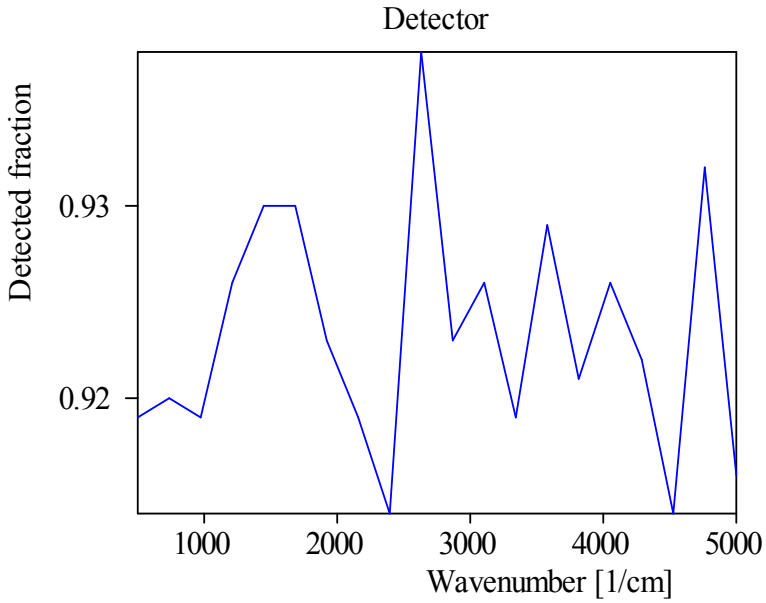
Since we want to be able to predict optical spectra we need to introduce a detector that records the amount of received rays for each spectral point. Open the geometric object list and create an object of type 'Rectangular detector'. Name it simply 'Detector'. Like a screen, this is a rectangle counting how often it is hit by a ray. However, instead of saving the position of the hit point, it records spectral information. The fraction of received rays is recorded for each simulated spectral point. Place the rectangle a little behind the screen and give it the same size and orientation. Make sure that the surface normal points to the side of the fiber. The following dialog shows the appropriate settings:



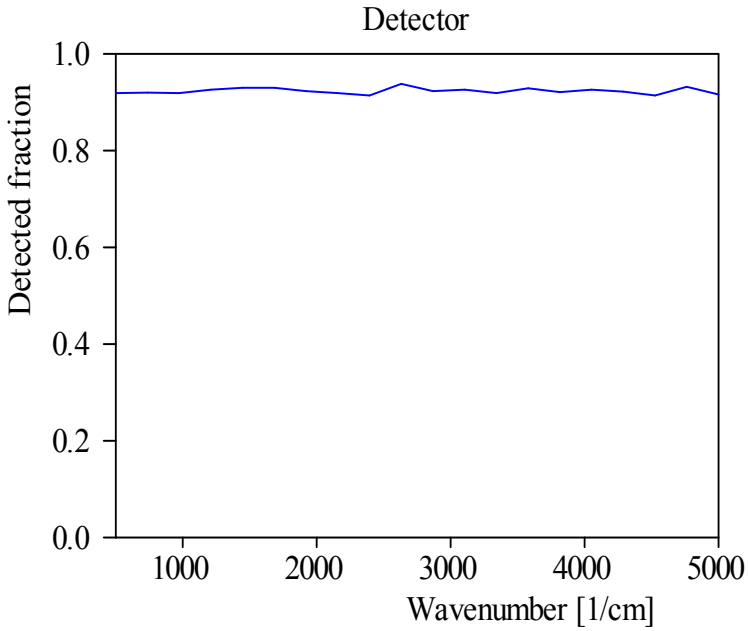
Open the rendered view 'Side' and update the picture by the **Draw** command to inspect the setup:



The detector is positioned to the right of the screen and will detect the rays passing the right end of the fiber and the screen. Now it is time to record the first SPRAY spectrum. In the main window, press the **Simulation** button. When the computation is finished, expand the 'Detector' treeview branch and right-click the subbranch 'Spectrum'. A new window opens displaying the recorded spectrum. Press the 'a' key on the keyboard to do an automatic scaling. The spectrum should be similar to this one:



Apply the command **Graphics|Edit plot parameters** and change in the following dialog the minimum and the maximum of the y-axis to 0.0 and 1.0, respectively. The parameter called 'Tick spacing' should be set to 0.2 . The graph changes to this:

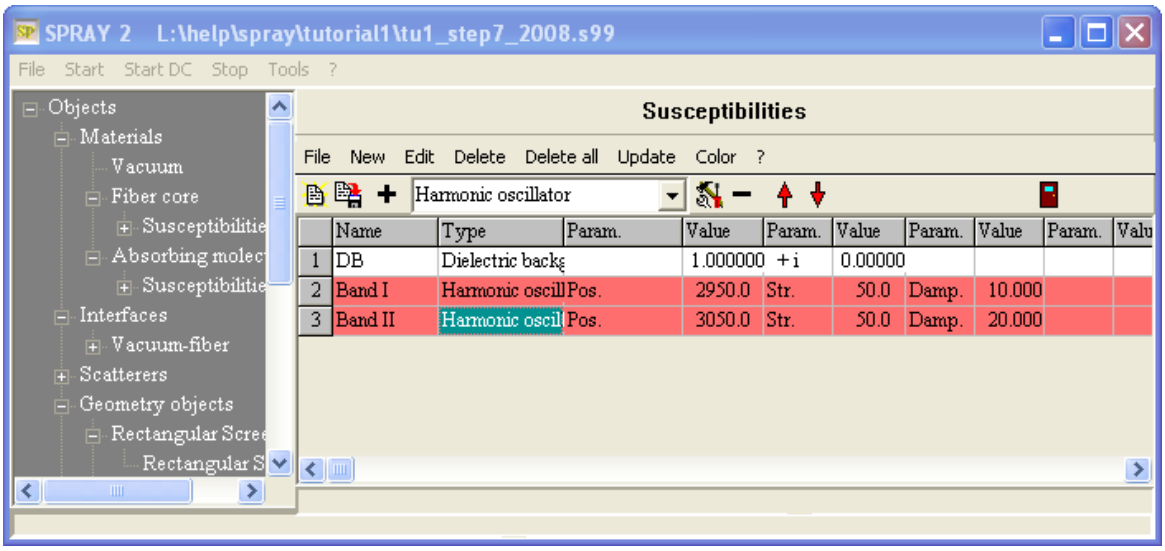


The spectrum does not show too much structure. In fact, it should be constant since no part of the present model contains any spectral variation. The variations in the spectrum are due to statistical noise. Remember, that we work with 1000 rays per spectral point. The expected noise has a level of about the square root of 1000 divided by 1000, which is - roughly - 3%. In the next step we will add an absorbing thin film to the model which will cause some more interesting features.

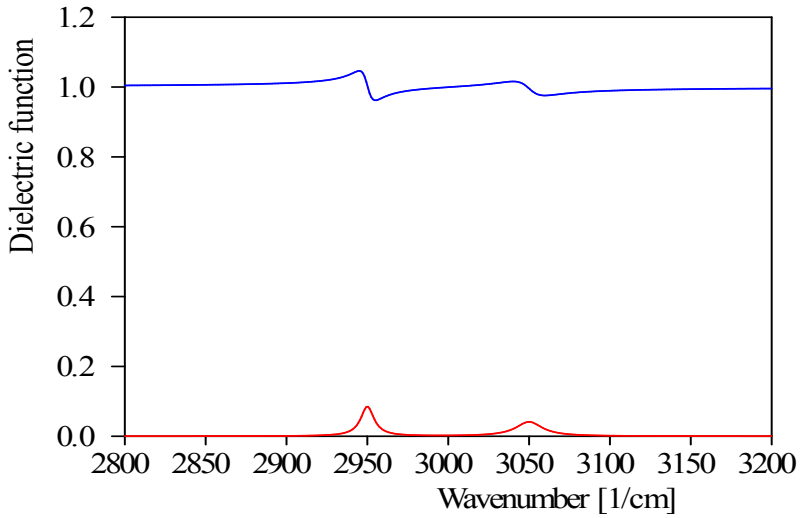
The configuration achieved up to now is saved as tu1_step7.s99.

2.6 Adding a coating

In the final step of this first tutorial we will add a thin absorbing film to the model. The film covers the fiber and leads to absorption bands in the detector spectrum.
First we have to create a new material that is going to represent the coating. Open the list of materials and create another entry of type 'Dielectric function model'. Name it 'Absorbing molecules'. Open the corresponding window and set a **Range** of 2800 ... 3200 1/cm with 400 data points. Open the list of **Susceptibilities**. Add a contribution of type 'Dielectric background' and two harmonic oscillators. Set the parameters of these susceptibilities to the ones shown in the following dialog:

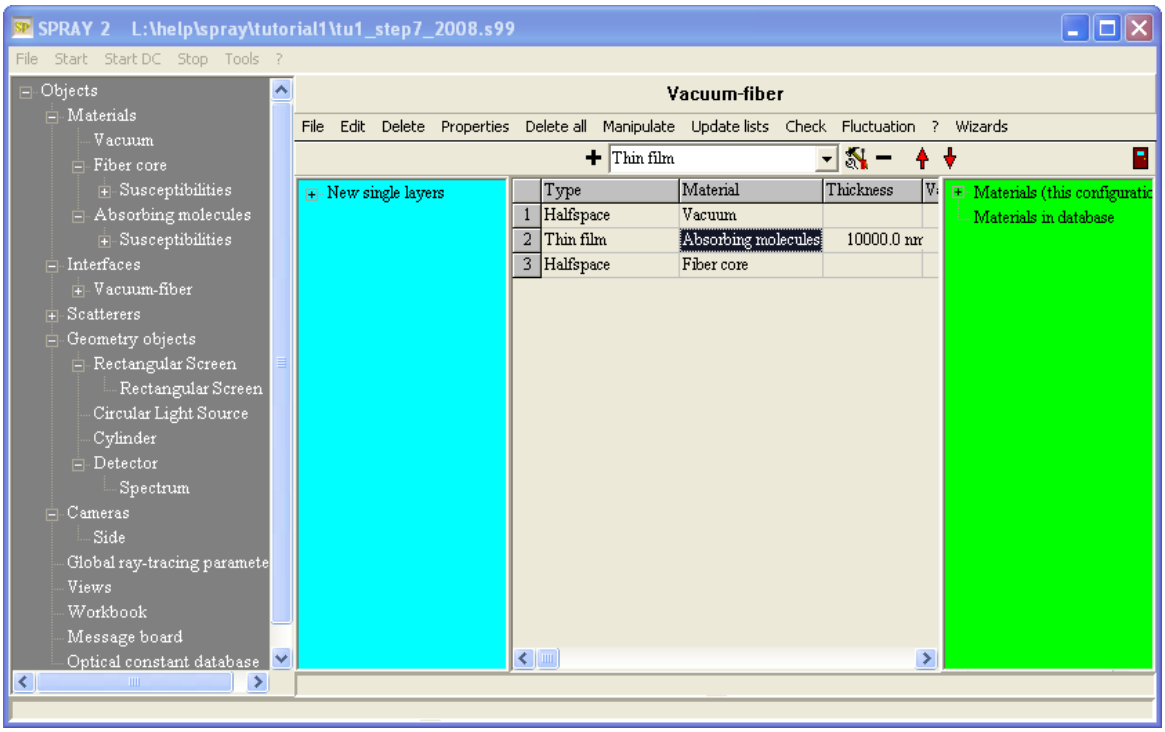


Close the list of susceptibilities and compute with the **Recalc** command the dielectric function called 'Absorbing molecules'. Use the command **Graphics|Real part plot parameters** to set the scaling of the plot similar to the following (see the graphics help to get information on the meaning of all the graphics parameters):

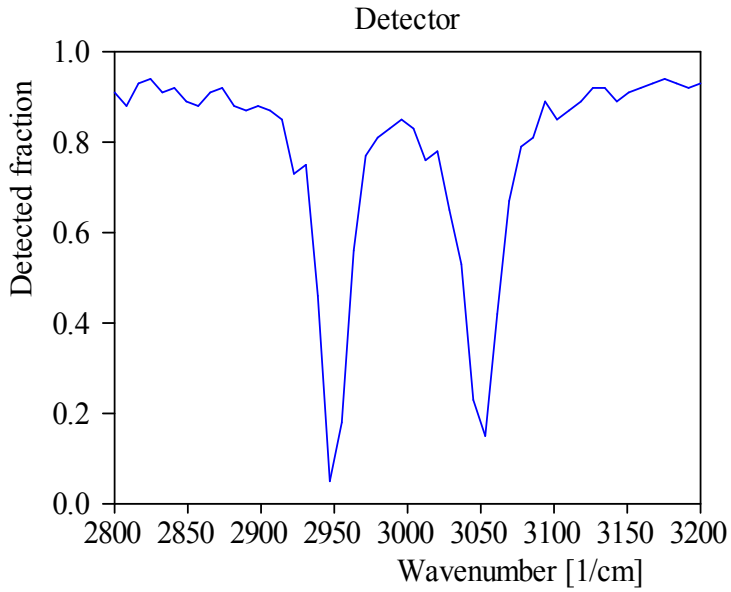


Now open the list of **Interfaces** and there the entry called 'Vacuum-fiber'. Place the cursor in line 2 (the one with the entry 'Fiber core'). Press the '+' button to create a new entry of type

'Thin film' in between the two halfspaces. Drag the 'Absorbing molecules' from the list of dielectric functions to the newly created layer. Set its thickness to 10000 (which means 10000 nanometers). The layer stack should now look like this:



In the treeview, opent the object **'Global ray-tracing parameters'** and set a spectral range from 2800 to 3200 1/cm for the spectral ray-tracing. Use 50 spectral points and 100 rays per spectral point for a quick first test. Then press **Start** in the main window. Open the window that displays the detector spectrum. Try to adjust the graphics parameters to achieve a picture like this:



If you could follow the tutorial up to this point: ***Congratulation!*** You master the basic features of SPRAY. If not, save your configuration and load the configuration tu1_step8.s99 which

contains the present status of the model. Inspect the settings and try to find the differences to your model in order to find out what's wrong with your parameters.

Part III

3 Light scattering: Computing diffuse reflectance spectra

3.1 Overview

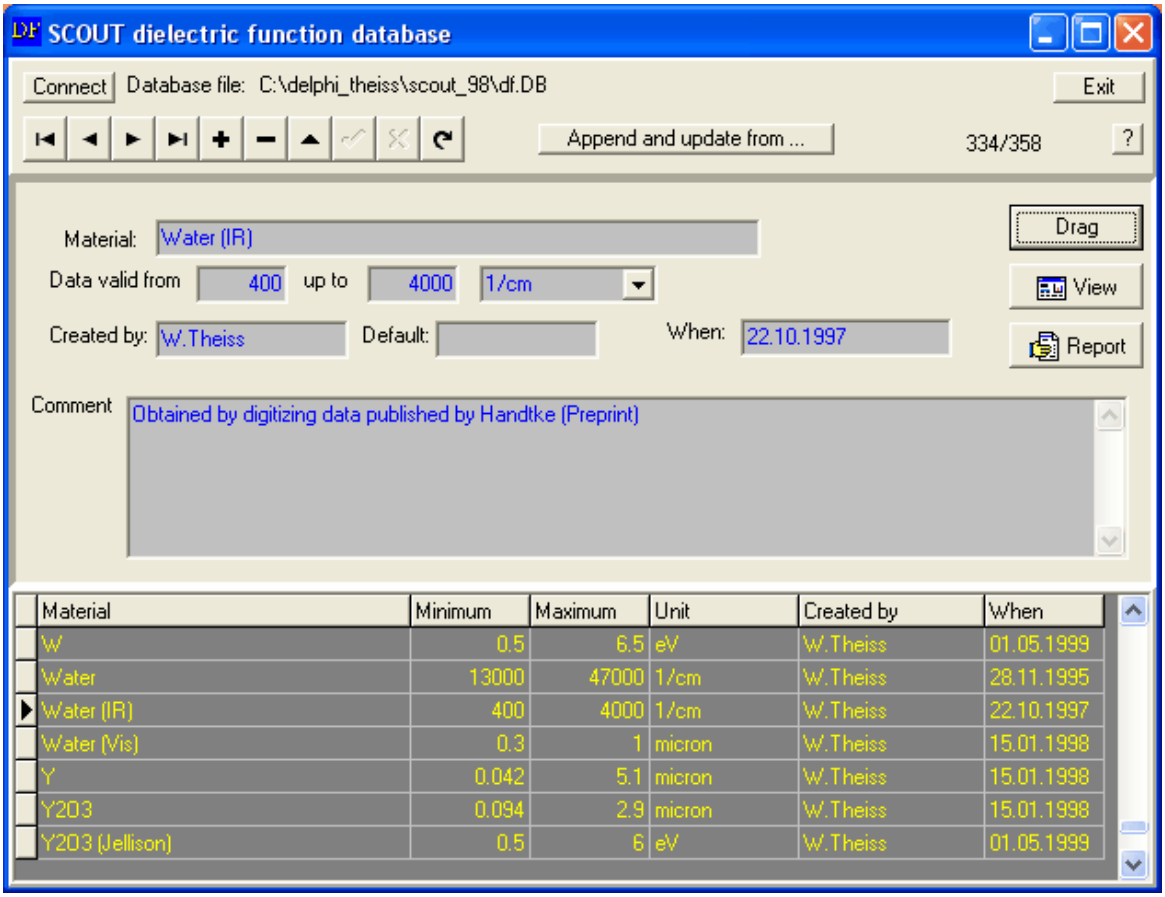
This tutorial shows you how SPRAY can be used to compute spectra of light scattering systems like powders, paints or colloidal suspensions. As an example we compute diffuse reflectance spectra in the infrared. The sample system is a collection of glass spheres (with and without a thin surrounding water film) in a metallic cup. Since the focus of this section is on the computation of scattering and absorption effects we use very simplified illumination and detection conditions. Much more realistic experimental setups for recording diffuse reflectance could be simulated in SPRAY as well, of course.

In the first section, we will import all required optical constants from the database and define the interfaces that will be used later on. Then the scattering and absorption properties of single (coated and uncoated) spheres are determined. After the definition of the sample geometry some diffuse reflectance spectra are computed for a number of configurations.

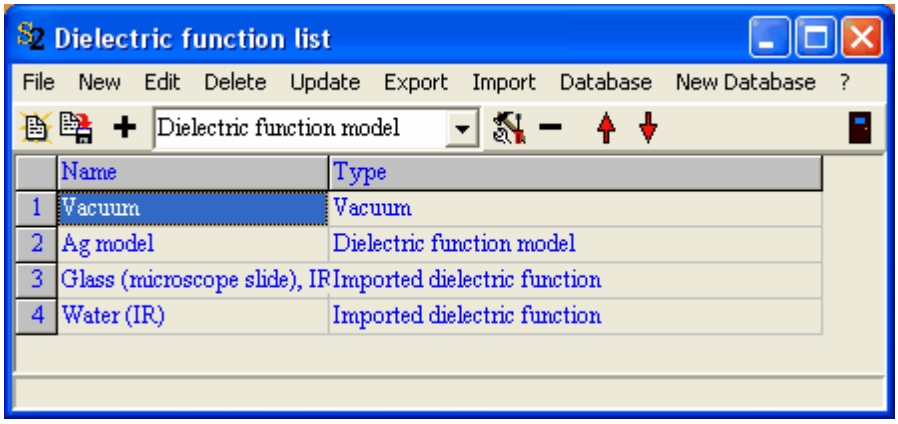
3.2 Defining the required interfaces

We will start with a complete empty SPRAY configuration and build up the final configuration step-by-step.

Database import
First the required optical constants have to be loaded from the database. Start the program and go to the list of optical constants by pressing the button '**Dielectric functions**' in the main window. Now use the **Database** command to open the database window:

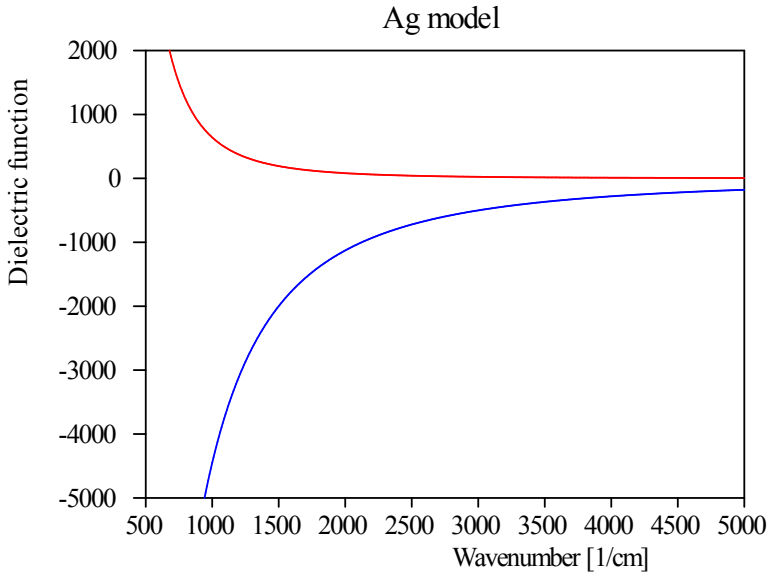


In order to work with a material from the database, you have to transfer it to the list of optical constants. Select the wanted database item (like 'Water (IR)' in the picture above), and then start a **drag&drop** operation from the button labeled **Drag** in the upper right corner. Move the mouse cursor into the list of dielectric functions and drop the optical constants here.
We need three materials for our computations: A metal for the sample cup, glass and water for the coated spheres. Please try to find the entries 'Ag model', 'Glass (microscope slide), IR' and 'Water (IR)' in the database and move them to the list of optical constants which should look like this:



Spectral range
Before we can use these materials for SPRAY computations we have to make sure that their

spectral range is defined properly. The final diffuse reflectance spectra will be computed in the range 500 ... 5000 1/cm. The range of the individual objects should cover this range (or be larger). Select the 'Ag model' entry and open its window with **Edit**. Since this is a model, we can compute required data in any spectral range. Use the **Range** command and select 500 ... 5000 1/cm with 451 data points (i.e. a 10 1/cm resolution). If you want to see the computed optical constants you can press the 'a' on the keyboard for automatic scaling. Eventually you have to change the graphics parameters a little bit in order to see the data like this (see the graphics course for details):

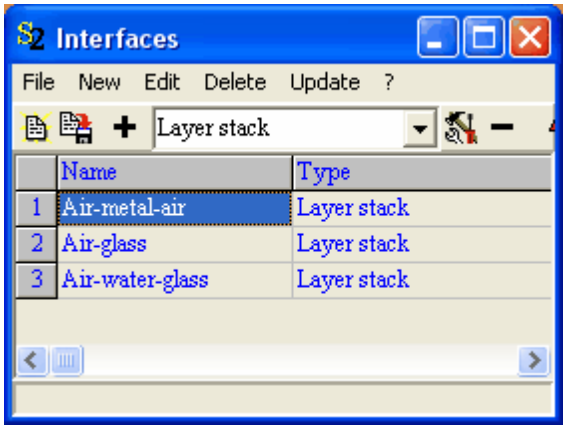


The data for glass and water are fixed imported data sets. Their spectral range cannot be changed. SPRAY will use linear interpolation to compute required values from the given data points.

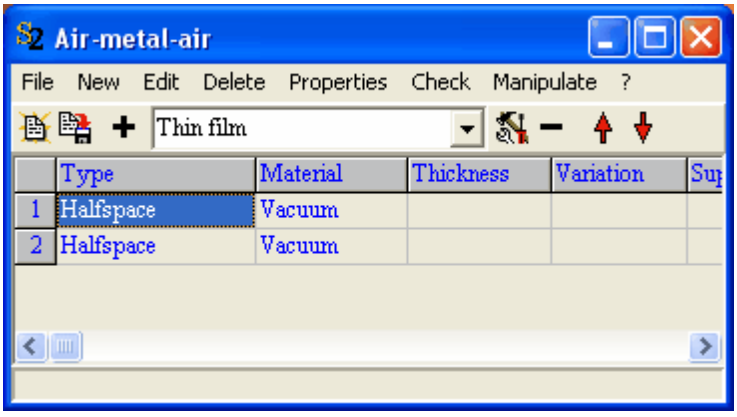
Interfaces

Now we can define the three interfaces we will need in the following: One for the sample cup walls, one for the uncoated air-glass transition and one for a wet glass surface, i.e. air-water-glass.

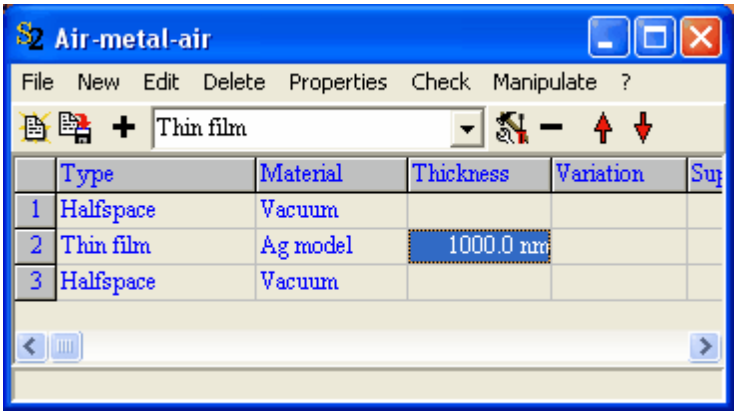
In the main window of SPRAY, click the **Interface** button to open the list of interfaces. Create three objects of type 'Layer stack' and name them like shown below:



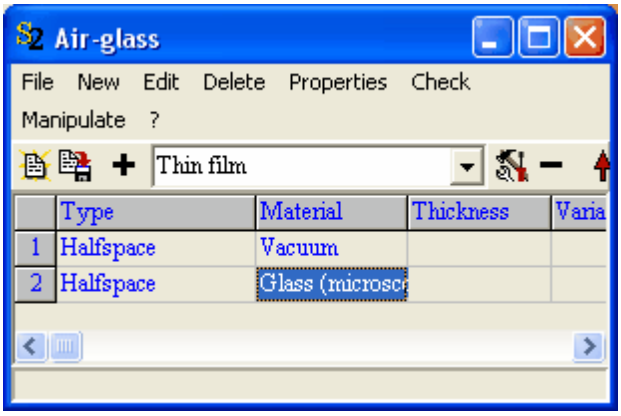
Select (like indicated above) the first layer stack and open its window with **Edit**:

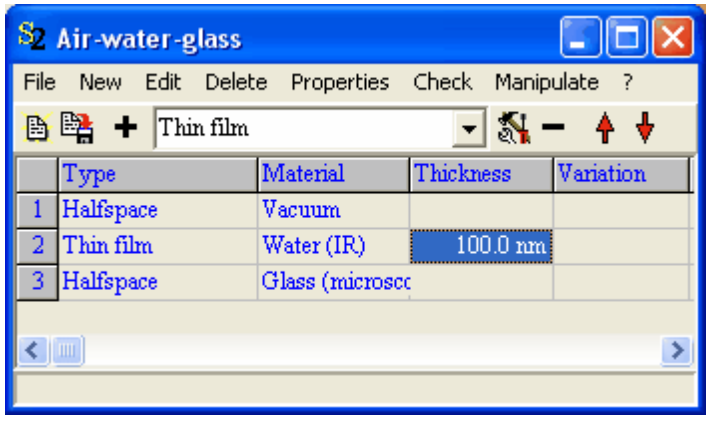


Select the bottom halfspace and press the '+' button to create a thin film in between the two halfspaces. Select the cell named 'Material' of the thin film and press the F4 function key to cycle through all possible choices for optical constants. Stop at 'Ag model'. Type in a new thickness of 1000 nm which will be thick enough to make this metal completely opaque in the infrared. The stack should now look like this:



Close this window and try to define the other two interfaces in a similar way. Here are the final stacks that you should try to reproduce:

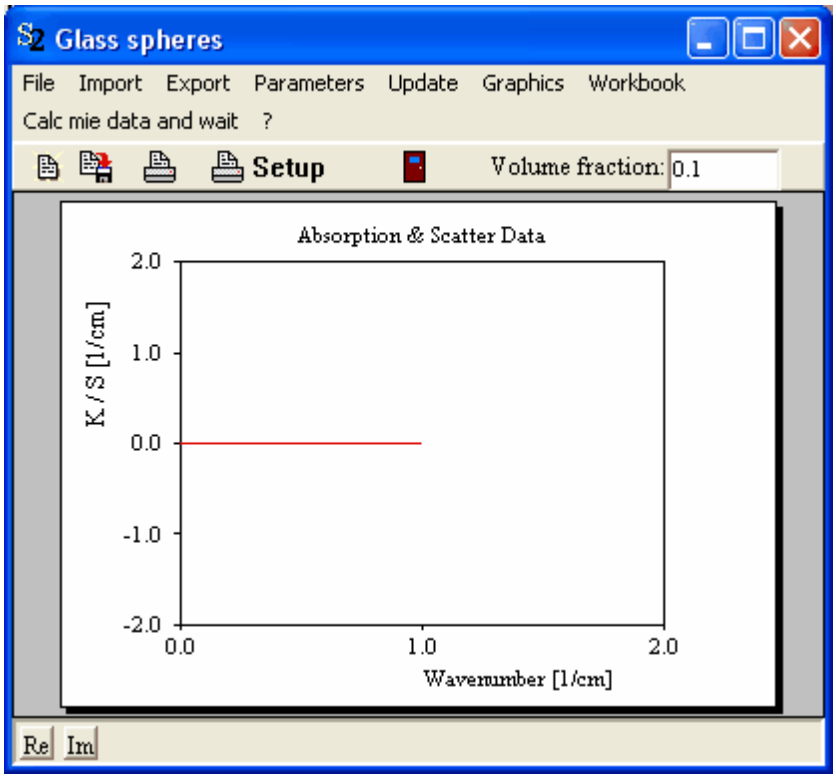




I hope you could follow this far. Otherwise you can load with **File|Open** the current configuration which I have saved under the name tu1_ex2_1.s99.

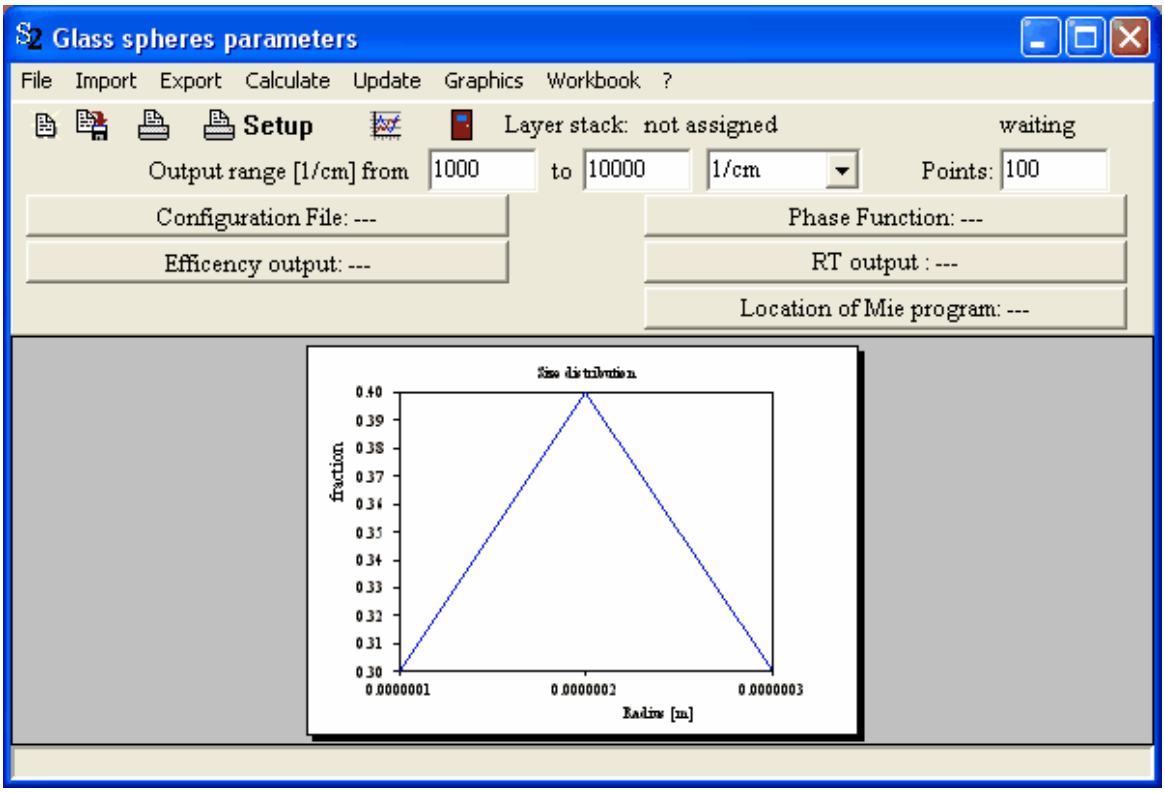
3.3 Compute single scattering characteristics

We are now ready to compute the single scattering properties of the glass spheres. This is done by an external program (Layeredsphere.exe) which is controlled by SPRAY as explained in detail in the SPRAY manual (section about 'Extended Mie scatterers'). Here I just quickly go through the necessary steps with some short comments.
Open the list of scatterers by clicking the button '**Scatterers**' in the main window of SPRAY. Create an item of type '**Extended Mie scatterer**', name it 'Glass spheres' and open it with the **Edit** command:



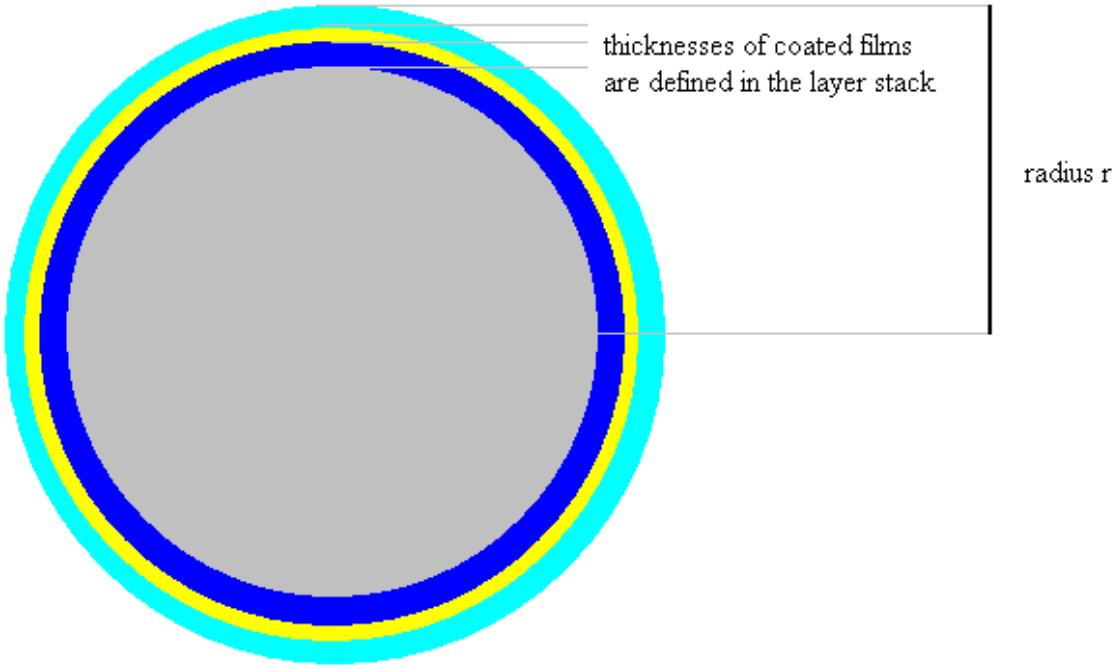
This window will show - after the Mie computation - the absorption and scattering coefficients of the scattering medium. You can change the volume fraction here without doing the single

scattering computations again. Just enter the new volume fraction (range: 0 ... 1.0) and press **Update** to re-compute the scattering and absorption coefficients.
However, once we have to do the single scattering computations. The required parameter settings are done in a separate window which is opened by the **Parameters** menu command:



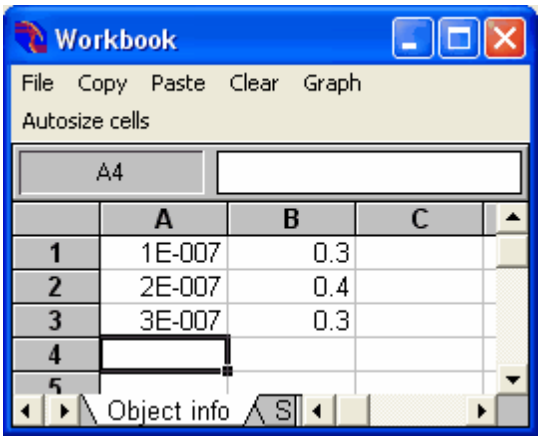
The following items have to be defined:

Layer stack
The computations are done for coated spheres like the one in the following sketch:

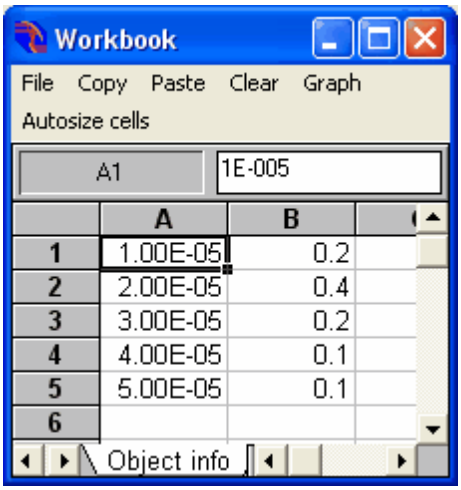


In the top of the window you can assign a layer stack. This is done by **Drag&Drop** from the list of interfaces to the label '**Layer stack: not assigned**'. The bottom halfspace of the layer stack represents the core material of the sphere, the top halfspace the outer world. If the stack contains thin films then these define the coating of the sphere.
For the first computation we use uncoated glass spheres and assign the interface 'Air-glass' (see above).

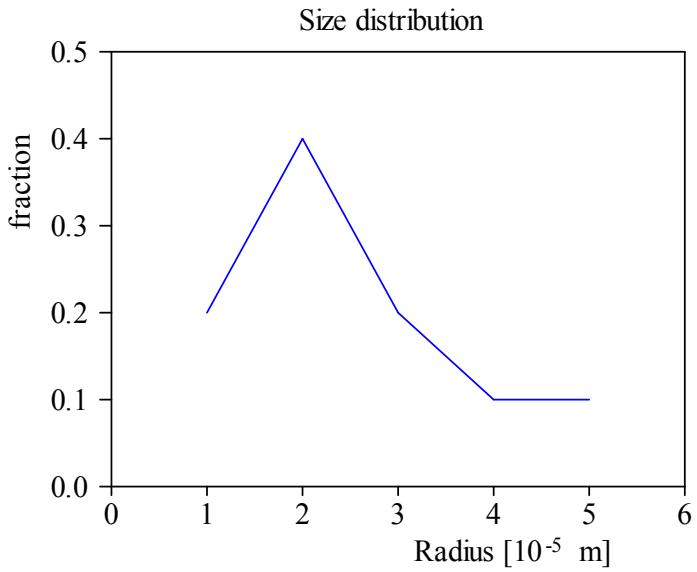
Radius distribution
The curve displayed in the parameters window defines the radius distributions. The sphere radius is the distance from the sphere center to the outer interface to the top halfspace (as indicated in the sketch above). The radius must therefor be larger than the sum of the thin film thicknesses, of course.
The easiest way to define the radius distribution is to use the workbook. Use the command **Workbook|Export xy** to write the current distribution to the workbook. Open the workbook with **Workbook|Open workbook**:



These numbers mean that there are three size classes of spheres: 100, 200 and 300 nm with probabilities 0.3, 0.4 and 0.3, respectively.
To enter your own distribution just modify the table as you like, place the cursor in the cell A1 and use the **Workbook|Import xy** command to import the new data as radius distribution. Eventually you should modify the graphics settings for an appropriate display of the new curve. I prepared the following distribution



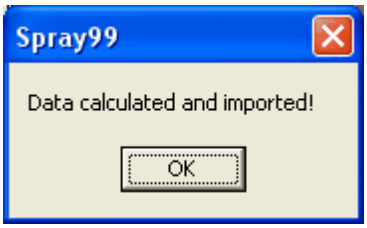
and obtained this picture:



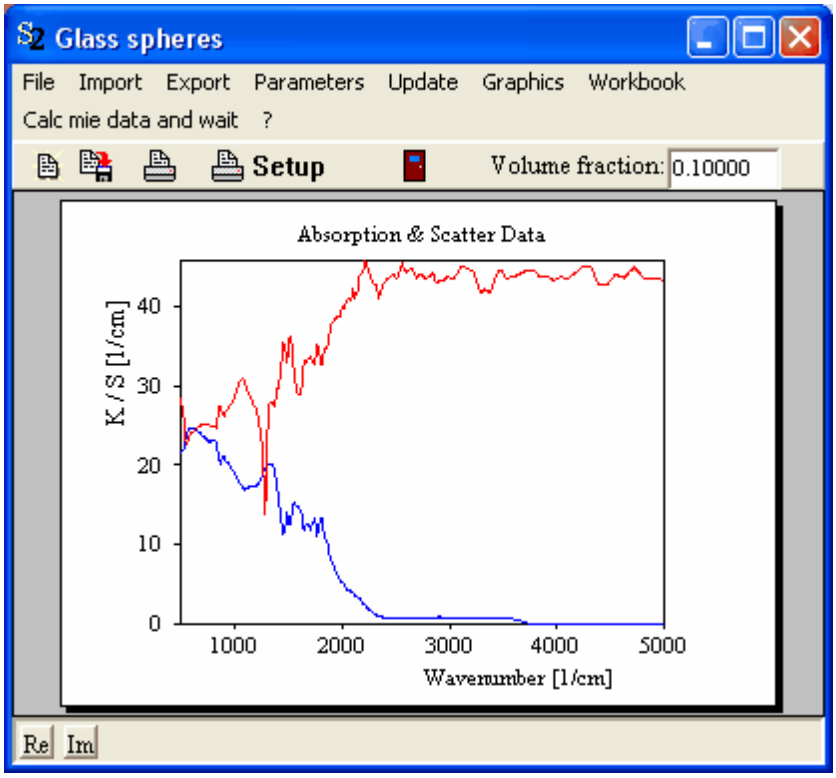
Output range
The output range of the Mie data should cover the range that will be used in the end to compute the diffuse reflectance spectra. In the simplest case you will just use the same settings.
On the other hand, if the Mie computations have to be done just once, you can do them with more points in order to be prepared for the case that you want to increase the spectral resolution of your diffuse reflectance spectra. The speed of your computer is also an important point in these considerations.
The following computations are done in the range 500 ... 5000 1/cm with 451 points.

Transfer filenames
We are almost ready now. In this final step we have to set several filenames. The corresponding files are created and used for the communication of SPRAY and the external Mie program. Please use the filename 'mie1' without file extension in the file dialogs that open after pressing the buttons **Configuration file**, **Efficiency output**, **Phase function** and **RT output**.
In the end you have to locate the Mie program 'Layeredsphere.exe' on your computer after pressing the button **Location of Mie program**.

Doing the Mie computation
Once you have done all settings you can close the parameters window and go back to the window showing the absorption and scattering coefficients. The menu command **Calc Mie data and wait** starts the Mie computations. Depending on the optical constants, the complexity of the coating, the size distribution, the output range and the speed of your computer the calculations can take quite some time.
For each size class of the radius distribution the external program is called once. This may result in some flashing on your screen since the program is running in a Dos box of Windows. SPRAY opens the following dialog to indicate the end of the numerical work:



The computed absorption and scattering coefficients are displayed as follows:

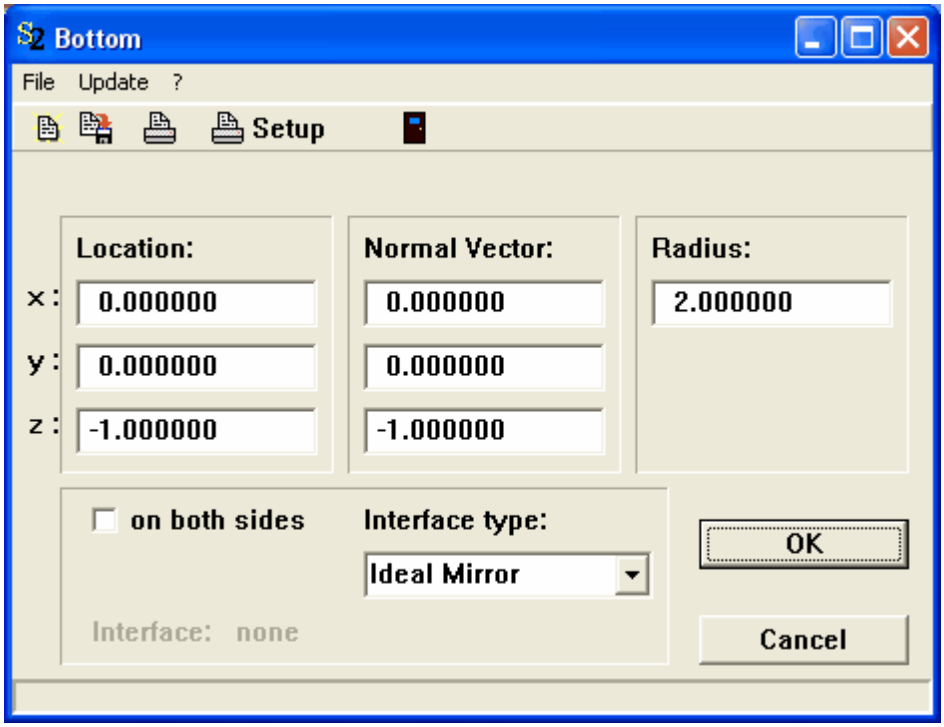


The configuration achieved up to now is stored under the name tu1_ex2_2.s99.

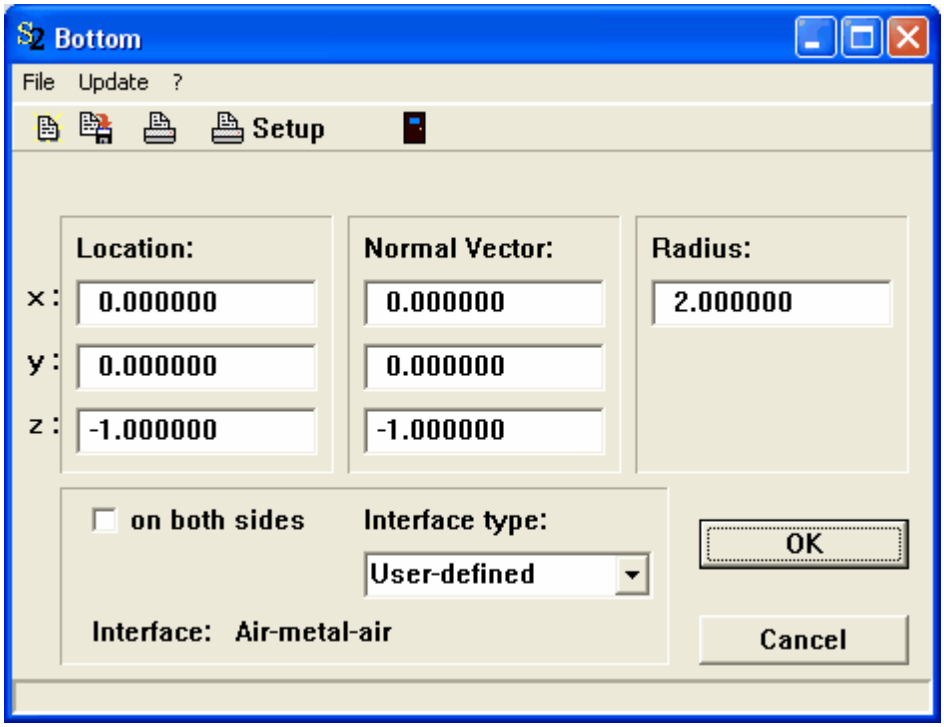
The angle dependence of the scattered light can be inspected with the View_RT utility program that comes with SPRAY. Please consult the SPRAY manual how to do this.

3.4 Setup of the sample system

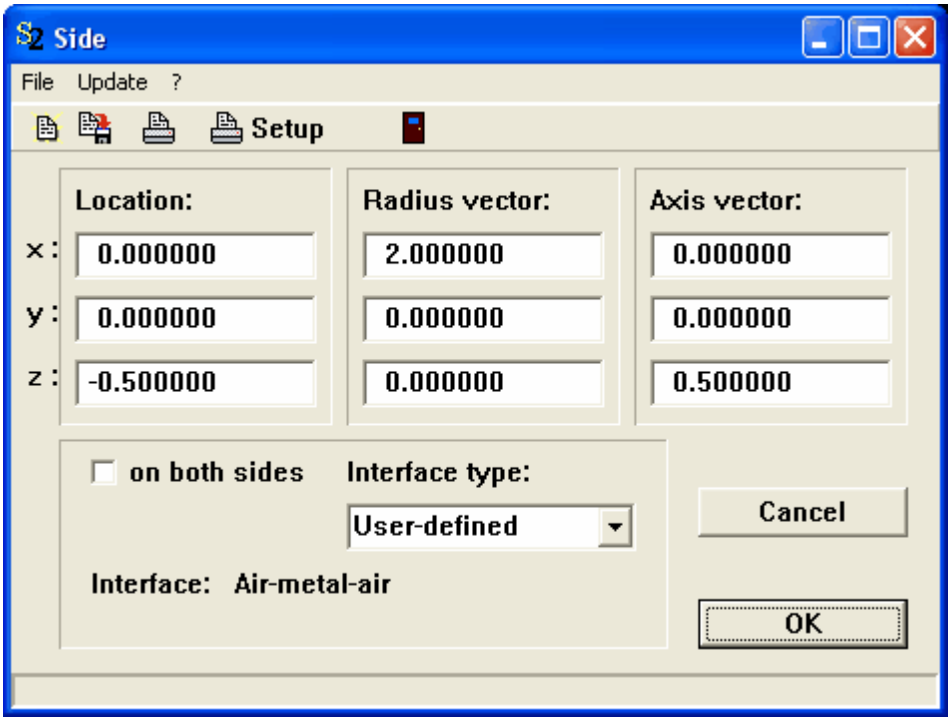
We will now setup a simple system for 'recording' the diffuse reflectance spectrum of the glass spheres.
We will put the glass spheres in a cup with metallic walls. The bottom of the cup is realized as a circular interface, the side walls are made using an open cylinder. Open the list of geometric objects by pressing the **Objects** button in the main window of SPRAY. Create a circular interface (object type 'Circle') with name 'Bottom' and the following coordinates:



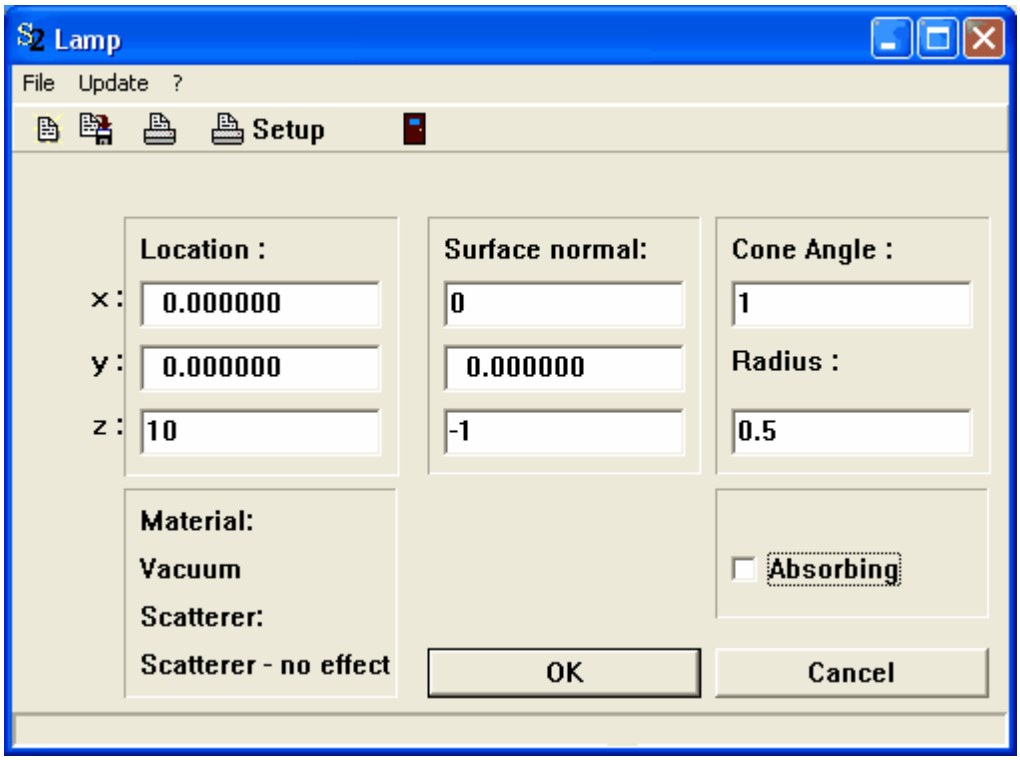
Note that the normal vector points downward. Now open the list of interfaces and drag the interface 'Air-metal-air' to the label 'Interface: None' and drop it there. If the dialog now looks like the following, press OK to close it:



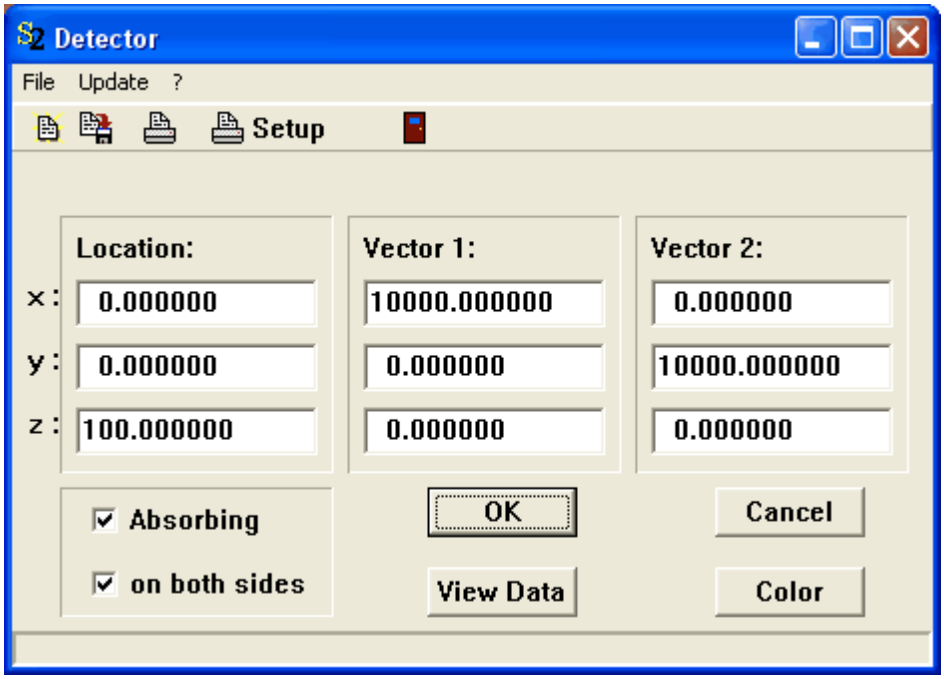
Now create an object of type 'Cylinder (open)', name it 'Side' and enter the following settings (do not forget to assign the interface by Drag&Drop from the list of interfaces):



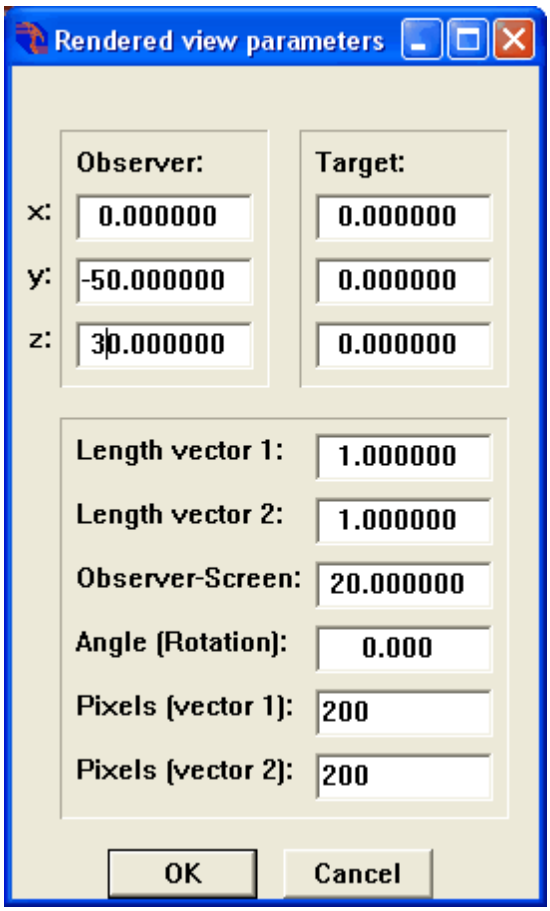
The illumination of the sample cup is done from above, using a light source of type 'Circular light source' (I named it 'Lamp') with these coordinates and assignments:



Note that the light source is not absorbing. We will finally put a huge rectangular detector above the present setup which will collect all rays that are reflected. Create an object of type 'Rectangular detector' and enter the these numbers:

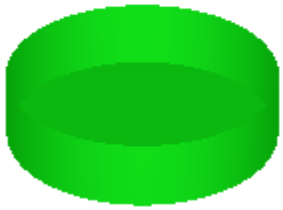


In the section about basic skills you have learned how to define 'rendered views' which should be used to check the setup. Create one with the following parameters:

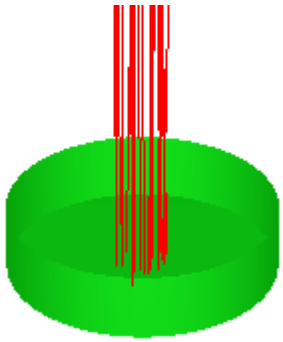


If everything worked out OK the rendered view will show the following image (Use the **Draw**

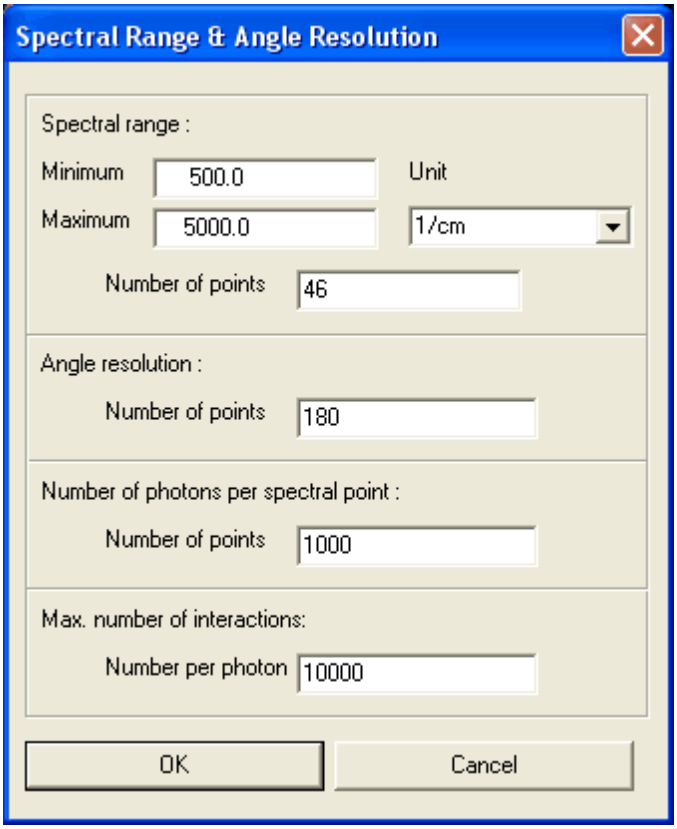
command):



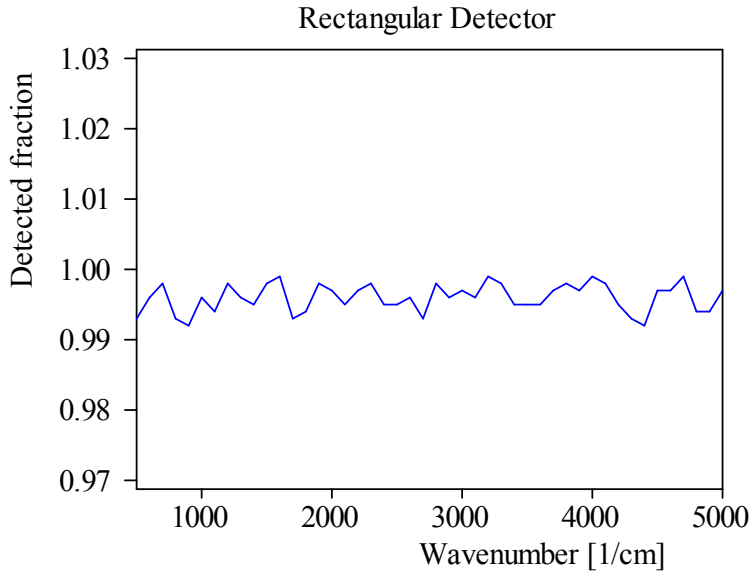
Sending 20 test rays (command **20**) you can see the reflection of the metallic bottom interface:



It is now time for a first test spectrum. Open the simulation parameters with **Parameters** in the main window. Copy the following simulation settings:

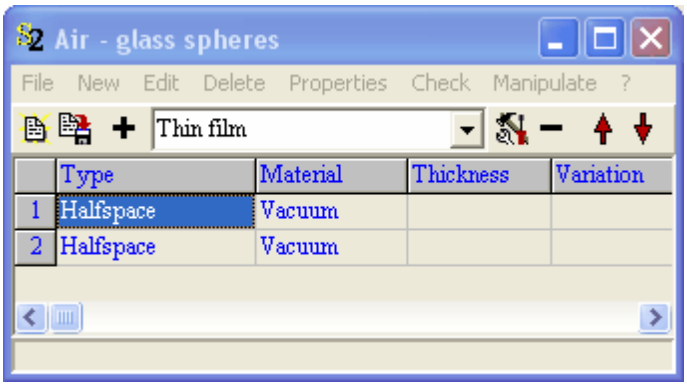


Then run the simulation by pressing the **Simulation** button in the main window. The simulation should not take very long because there is just a simple reflection of the rays at the bottom interface. Since silver is a noble metal its reflectivity in the infrared is very high. Open the detector object in the list of objects and use the **View data** command to display the spectrum. I obtained the following spectrum which confirms the expected result:

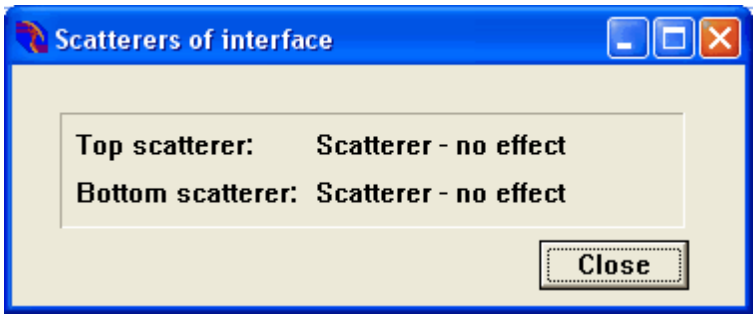


What is left to do is to add the scattering glass spheres to the system. This requires some further

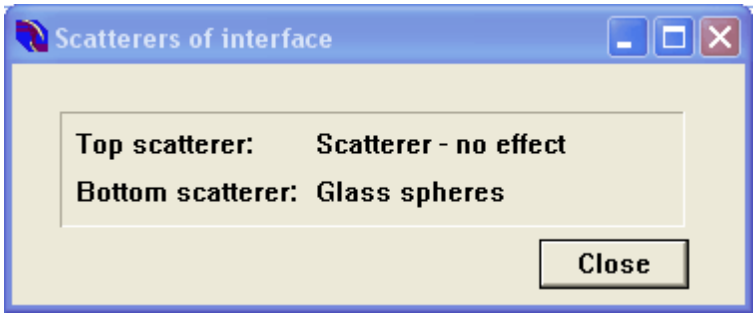
actions. First we need another interface which defines the transition from air (filled with no scatterers) to air filled with our scattering glass spheres. To do this open the list of interface again and create another layer stack, named 'Air - glass spheres'. Open the layer stack definition:



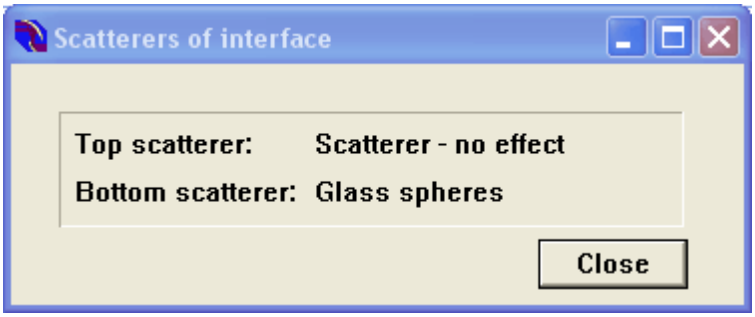
Since the glass spheres are embedded in air, there is no real physical transition from one material to another, and vacuum is used on both sides of the interface. However, crossing the interface means to enter a scattering medium. You can set the scatterers on both sides of the interface by using the **Properties|scatterers** command. The following dialog opens:



You can assign scatterers to the top and bottom halfspace by Drag&Drop from the list of scatterers. In this case, we need the following assignment:

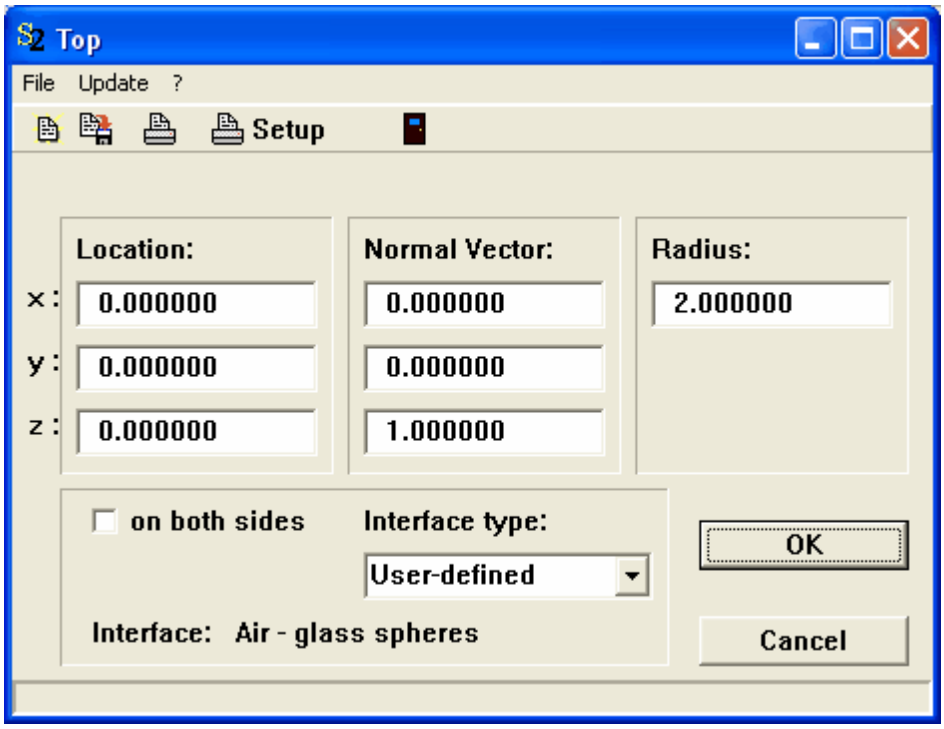


Also the interface 'Air-metal-air' has to know which scatterers are on which side. Open this interface from the list of interfaces, use the **Properties|scatterers** command and do the same assignment as above:

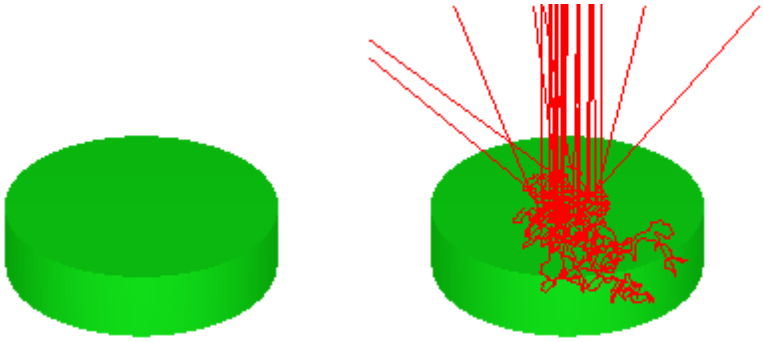


This means that there are scattering glass spheres inside the cylinder and no scattering objects outside.

The very last action is to introduce another circular interface at the top of the sample cup which defines where in space the scattering medium begins. Go to the list of objects, create a new object of type 'Circle' and assign the interface 'Air - glass spheres' to it. The geometrical parameters are the following:



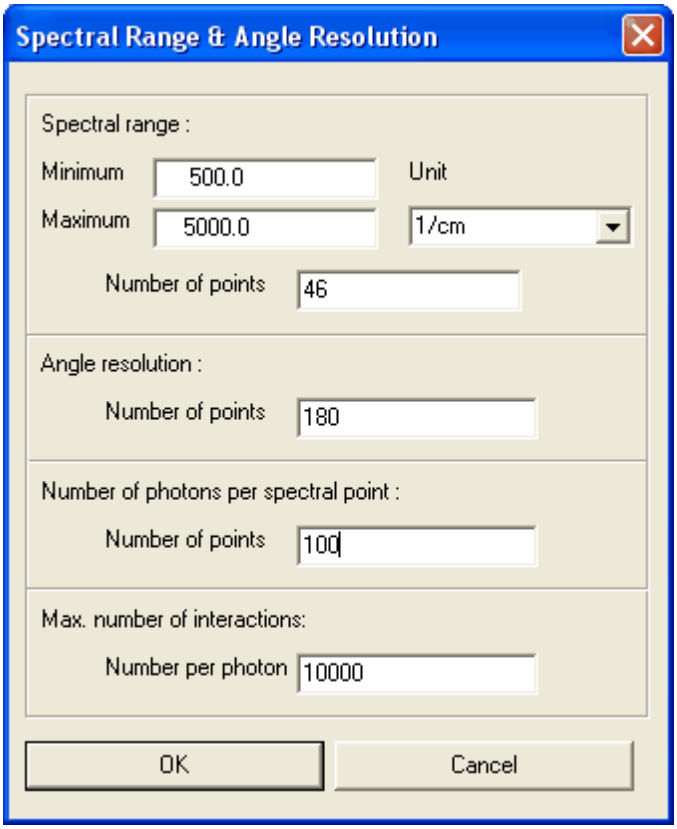
These modifications should be checked in the view object. Open it and repaint the image with **Draw**. Sending some test rays clearly shows the diffuse light scattering now:



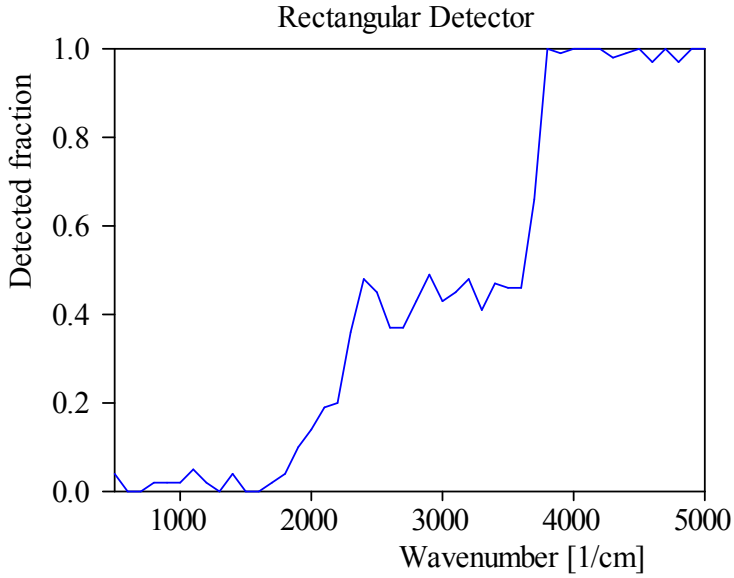
That's it! We are now ready to compute the first diffuse reflectance spectrum of scattering glass spheres in a metallic cup. The current configuration has been saved in the file tu1_ex2_3.s99.

3.5 Results

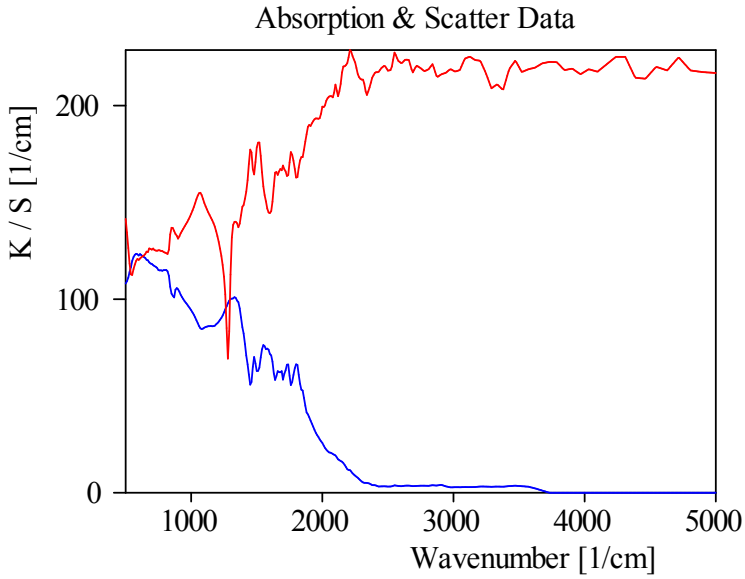
In order to compute a first test spectrum, go to the main window and set the simulation parameters as shown below (Click the Parameters button to open this dialog):



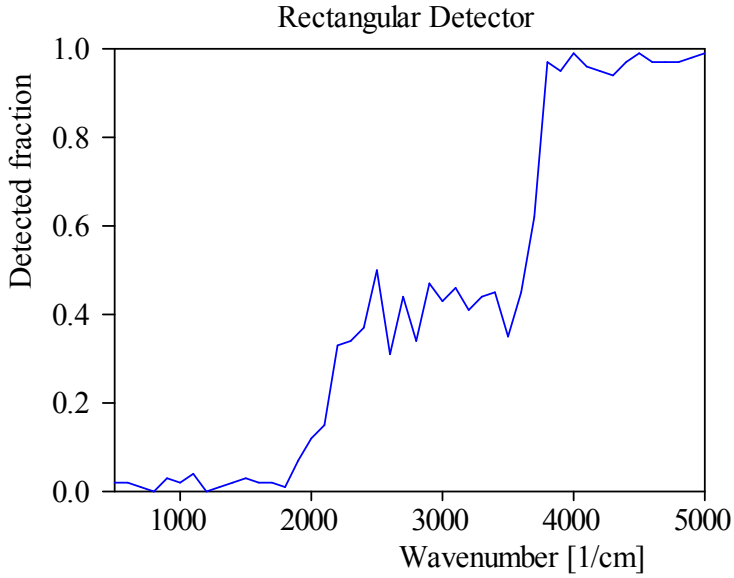
Then press the **Simulation** button to start the ray-tracing work. After a while you can inspect the result: Open the list of objects and there the detector object. Display the spectrum with **View data** (Eventually the graphics settings must be modified):



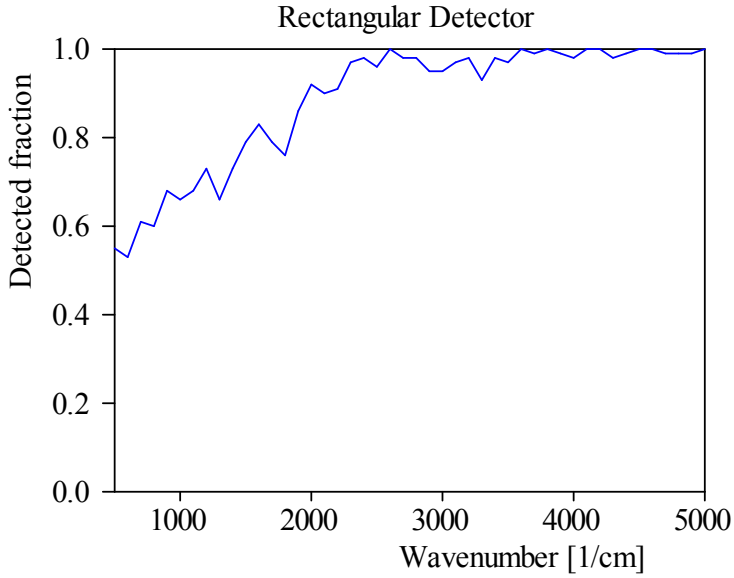
Note that the volume fraction of 0.1 for the glass spheres is not very realistic since they are not embedded in a solid host material. Go to the list of scatterers, open the 'Glass spheres' and set the volume fraction to a more realistic value of 0.5 . Use the **Update** command to recompute the scattering and absorption coefficients:



Now do the simulation again. The result

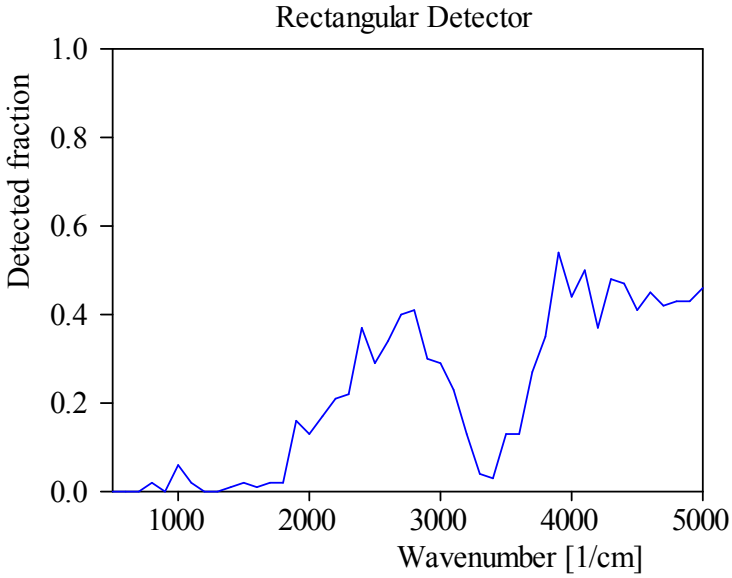


is not very different from the case above. This is due to the fact that almost no ray reaches the side walls in both cases, and what you see is a 'halfspace spectrum' of scattering glass spheres in both cases.
Not in reality, but in the model we can set the glass sphere volume fraction to much lower values, for example to 0.001, and recompute the diffuse reflectance spectrum:



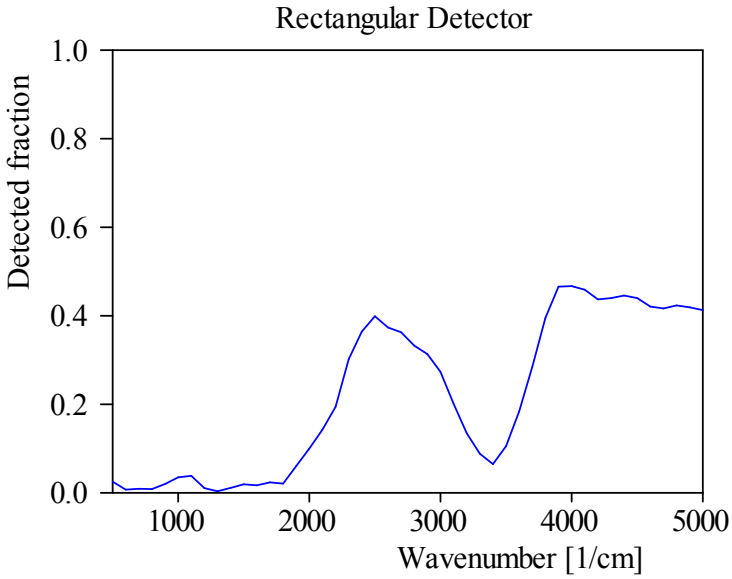
Now the spectrum is dominated by the large regular reflection of radiation by the metal on the bottom of the cup. The diluted glass spheres absorb significantly below 2000 1/cm only.
It's a good exercise for you to place some screens and detectors in the system to investigate where is how much radiation. Especially in the 'dense' system with a volume fraction of 0.5 it might be of interest to find out how deep the radiation penetrates the scattering system, or how far from the incident beam the light comes out again.

To finish this tutorial we now compute the diffuse reflectance spectrum for the water coated glass spheres. Open the list of interfaces and the list of scatterers. Open the scatterers called 'Glass spheres' and there the parameters window. Assign the layer stack 'Air-water-glass' to the sphere coating, and use the filename mie2 for the transfer and output files. Then close the parameters window and use the command **Calc Mie data and wait** to recompute the scattering and absorption characteristics. Set the volume fraction back to 0.5 and do the ray-tracing simulation again. The detector signal (which is the diffuse reflectance of the sample) is much lower now due to water absorption:



This configuration can be found in tu1_ex2_4.s99.

Serious results should be obtained with a much lower noise level, of course. In order to show the difference I have computed the spectrum above again, now with 10000 rays per spectral point instead of 100. Here is the result:



Index

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -

- V -

- W -