How to Work with the Programmable Spectrum and Example (Black-Body Radiation)
Abstract

Providing maximum versatility for your optical simulations is one of our most fundamental objectives. In this document we show you how to work with the Programmable Spectrum: that is, how to define a function that assigns a different complex weight to each wavelength/frequency present in the spectral make-up of a field, working under assumptions of stationary behaviour. The black-body emitter is one of the default spectrum models in VirtualLab, but we use it here as a basic programming example.
Where to Find the Programmable Spectrum

1. Navigate to the Programmable Spectrum section.
2. Click on the 'Edit' button to open the settings editor.
3. Customize the settings as needed.
4. Use the Source Code Editor to write your code.
5. Review and save your changes.
Setting Up the Sampling

Note that you have the option to use either wavelength or frequency as your independent variable!
Writing the Code

The panel on the right shows a list of available independent parameters.

**Position** represents the independent variable (either wavelength or frequency, as pre-set in the configuration dialogue).

The code in the Main Function must return a `Complex` value per **Position**, which is determined by the function programmed by the user.

Use the Snippet Body to group parts of the code in support functions.

The final sampling of the function is determined by the settings from the previous dialogue.

---

**Hint:** The Global Parameters, Snippet Help, Advanced Settings tabs and other aspects of the interface work equivalently to those of other programmable elements in VirtualLab.
Output

• The output is a one-dimensional graph of the programmed complex-valued function.

• It is possible to use the generated spectrum as the spectral make-up of the source in your Optical Setup.

• The number of separate spectral modes when the programmed spectrum is used in a source is equivalent to the number of samples in the spectrum.
Programming a Black-Body Spectrum
Black-Body Radiation

The power density associated to each wavelength (spectral density) when an emitter is assumed to behave like a black body at a certain temperature $T$ is given by Planck’s Law:

$$S(\lambda) = \frac{8\pi hc}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda kT}\right) - 1} \tag{1}$$

$S(\lambda)$ → Spectral density  
$\lambda$ → Wavelength  
$h$ → Planck’s constant  
$c$ → Speed of light in vacuum  
$k$ → Boltzmann constant  
$T$ → Absolute temperature of black-body emitter
Black-Body Radiation

The maximum of the curve is achieved for the wavelength

$$\lambda_{\text{max}} = \frac{b}{T} \quad (2)$$

where $b = 2.897\,772\,9 \times 10^{-3} \text{ m K}$ represents Wien’s displacement constant.
Programmable Spectrum: Setting Up the Sampling

- Specification in Wavelength Domain
- Central Wavelength = 600 nm
- Size of Wavelength Window = 1 um
- Number of Data Points = 200

Hint: Note that the Number of Data Points can affect simulation time/accuracy if spectrum is used in an actual source!
Programmable Spectrum: Entering the Programming Interface
Programmable Spectrum: Global Parameters

• Once you have triggered open the Edit dialogue, go to the Global Parameters tab.
• There, Add and Edit two global parameters:
  - `double TemperatureKelvin (0, NaN)`: represents the absolute temperature at which the black body is radiating.
  - `bool Normalize`: will the function be scaled so that the maximum allowed amplitude value is 1 (`true`) or not (`false`)?

*Hint:* it is possible to add some clarifying text to each global parameter to facilitate use of the snippet for other users!
Programmable Spectrum: Snippet Help

• **Optional:** you can use the Snippet Help tab to write instructions, clarifications, and some metadata associated to your snippet.

• This option is very helpful to keep track of your progress with a programmable element.

• It is especially useful when the programmable element is later disseminated to be handled by other users!

**Hint:** Use HTML commands to format the text.
Black-Body Spectrum

Version: 1.0
Last Modified: Tuesday, August 14, 2018

Black-body radiation is one of the best known theories in physics, and it is employed across several fields. This snippet generates a spectrum according to the black-body curve. The user can input which temperature the black body shall radiate at, and whether the spectrum should be scaled so that the maximum weight is unity.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TemperatureKelvin</td>
<td>The temperature, in Kelvin, of the black-body whose radiation is simulated by the spectrum generated with this snippet.</td>
</tr>
<tr>
<td>Normalize</td>
<td>This variable gives the user the option to scale the curve so that the maximum weight is unity.</td>
</tr>
</tbody>
</table>
Programmable Spectrum: Writing the Code

Declaration of output variable given by default

Code to run if curve is to be normalized

Code to run if curve is not to be normalized

Are there errors in your code?

Export Snippet to save your work!

Default global parameters/variables

Global parameters defined by user in Global Parameters tab

The normalization constant corresponds to the inverse of the value of the function at the extremum—that way, we ensure that the function will be scaled so that its maximum value will be one

Planck's Law is in dimensions of energy, and the spectrum definition in VL works with amplitudes!

Eq. (1)—normalized

Eq. (2)

Definition of physical constants not included in the default Globals class

The normalization constant corresponds to the inverse of the value of the function at the extremum—that way, we ensure that the function will be scaled so that its maximum value will be one

Planck's Law is in dimensions of energy, and the spectrum definition in VL works with amplitudes!

Eq. (2)

Declaration of output variable given by default

Code to run if curve is to be normalized

Code to run if curve is not to be normalized

Are there errors in your code?

Export Snippet to save your work!
Programmable Spectrum: Using Your Snippet

- You can modify the value of the global parameters you defined here.
- Modify the sampling parameters according to the requirements of your simulation.
- Bear in mind that the function we have programmed only works for wavelength specification!
- Modify your snippet again by clicking on Edit.
Programmable Spectrum: Output
Test the Code!

Complex value = new Complex(0, 0);

// Constants not included in Globals.
const double BoltzmannConstant = 1.3806505e-23;
const double ProportionalityConstantWienLaw = 2.8977729e-3;

if (Normalize) // Code to run if the curve is to be normalized.
{
    // Eq. (2) computes the wavelength at which the curve presents its maximum.
    double wavelengthMaximum = ProportionalityConstantWienLaw / TemperatureKelvin;
    // The normalization constant is equal to the value of the curve at wavelengthMaximum.
    double normalizationConstant = (Math.Pow(wavelengthMaximum, 5) * (Math.Exp((Globals.PlanckConstant * 
       Globals.VacuumSpeedOfLight) / (wavelengthMaximum * BoltzmannConstant * TemperatureKelvin)) - 1)) / 
        (8 * Math.PI * Globals.PlanckConstant *Globals.VacuumSpeedOfLight);
    // Eq. (1) multiplied by normalization constant gives the final value of S per wavelength.
    value = normalizationConstant * (8 * Math.PI * Globals.PlanckConstant *Globals.VacuumSpeedOfLight) / 
        * BoltzmannConstant * TemperatureKelvin)) - 1));
}

// Continued in next page.
// Continued from previous page.

else // Code to run if curve is not to be normalized.
{
    // Eq. (1) gives the value of S per wavelength.
    value = (8 * Math.PI * Globals.PlanckConstant * Globals.VacuumSpeedOfLight) /
                            (Position * BoltzmannConstant * TemperatureKelvin)) - 1));

    // Eq. (1) is in dimensions of energy, and the programmable spectrum in VirtualLab must return field
    // amplitudes:
    value = Complex.Sqrt(value);

    return value;

    // End of code.
How to Use Your Custom Spectrum in a Source
## Document Information

<table>
<thead>
<tr>
<th>title</th>
<th>How to Work with the Programmable Spectrum and Example (Black-Body Radiation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>document code</td>
<td>CZT.0095</td>
</tr>
<tr>
<td>version</td>
<td>1.0</td>
</tr>
<tr>
<td>toolbox(es)</td>
<td>Starter Toolbox</td>
</tr>
<tr>
<td>VL version used for simulations</td>
<td>7.4.0.49</td>
</tr>
<tr>
<td>category</td>
<td>Feature Use Case</td>
</tr>
<tr>
<td>further reading</td>
<td>- How to Work with the Programmable Light Source And Example (Gaussian Beam)</td>
</tr>
<tr>
<td></td>
<td>- Programming a Chirped Gaussian Pulse Spectrum</td>
</tr>
</tbody>
</table>