

# 3

## SIMULATION OF STIMULATED BRILLOUIN SCATTERING

### 3.1 INTRODUCTION

This chapter presents in detail the synthesis of the distributed Brillouin scattering modeling in optical fibers using a newly developed algorithm. The analyses of backscattered Brillouin signal under various operating parameters along the optical fibers are also presented using the developed simulation MATLAB codes.

### 3.2 MATLAB SIMULATIONS

There are principally two situations in simulating stimulated Brillouin scattering in optical fibers, namely, the Initially Values Problems (IVP) and the Boundary Values Problems (BVP). These two problems require different approaches in order to be solved, which are explained in the following sections.

#### 3.2.1 Initial Values Problems of Stimulated Brillouin Scattering

MATLAB has the capability to solve Ordinary Differential Equations (ODEs). In order to execute this task, a function file (odefilename.m) should have the following form:

```
function f = odefilename(t,y)
f = < Insert a function of t and/or y here >;
```

The function file should begin with “function”. Odefilename can be any name, but must be the same as the name of the file that will be saved. This file must be saved to the MATLAB search path. The intensities of the pump and stokes signals can both be considered as a single vector,  $I$  with  $I = \begin{bmatrix} I_p \\ I_s \end{bmatrix}$ . Then, in order to define the stimulated Brillouin scattering system, the ordinary differential equations given in equation Error! Reference source not found. and Error! Reference source not found. can be rewritten as the functions in vector form:

$$\frac{dI_p}{dz} = I_p' = I(1)' = -g_B I(1)I(2) - \alpha I(1) \quad (3.1)$$

$$\frac{dI_s}{dz} = I_s' = I(2)' = -g_B I(1)I(2) + \alpha I(2) \quad (3.2)$$

In MATLAB, a function for the above ordinary differential equations can be defined as follows:

```
function dIdz = odeSBS (z,I)
% I(1)= Ip
% I(2)= Is
alpha_dB = 0.2; %0 dB/km
alpha = 0.2/4.343; % /m
Gb = 5e-11; % m/W
dIdz(1) = - Gb*I(1)*I(2) - alpha*I(1);
dIdz(2) = - Gb*I(1)*I(2) + alpha*I(2);
dIdz = dIdz(:);
```