

Method of lines solution to the transient SBS equations for nanosecond Stokes pulses

F. S. Gökhan
fsgokhan@gmail.com

G. W. Griffiths
graham@griffiths1.com

W. E. Schiesser
west@Lehigh.EDU

Department of Electrical and Electronics Engineering, Havaalanı Yolu üzeri 8. km, Hasan Kalyoncu University, Sahinbey, Gaziantep, Turkey

School of Engineering and Mathematical Sciences, City University, London, UK

Engineering and Mathematics, Lehigh University, Bethlehem, PA, USA

The spectral and temporal evolution of distributed sensing based on stimulated Brillouin scattering (SBS) in optical fibers for several-nanosecond Stokes pulses is demonstrated by using the method of lines (MOL) solution of the transient SBS equations. A *superbee flux limiter* is utilized to avoid numerical damping and dispersion that would otherwise be brought on by the approximation of spatial derivatives associated with the partial differential equations (PDEs). In order to increase computational efficiency, an approach is adopted whereby the sparse PDE Jacobian matrix integrator option of the ODE solver(s) is employed. Simulation examples of SBS-based sensing for fibers containing sections with different Brillouin frequencies are presented. To the best of our knowledge, this MOL solution is proposed for the first time for modeling of the transient SBS equations for nanosecond Stokes pulses with different waveforms in a SBS based fiber optic sensor. [DOI: <http://dx.doi.org/10.2971/jeos.2013.13049>]

Keywords: Brillouin scattering, method of lines, finite differences, simulation, flux limiter, sparse matrix integrator

1 INTRODUCTION

Stimulated Brillouin scattering (SBS) in optical fibers permits us to measure temperature and/or strain on a truly distributed basis, over kilometric ranges with high resolution [1]– [3]. It consists in the coupling between two counter-propagating optical waves, the pump and the Stokes wave, and an acoustic wave. In this three-wave mixing process, power is transferred from the pump light wave to the Stokes light wave (that is, to the light wave having a lower frequency) and also to the acoustic wave. The interaction strength depends on the frequency offset between the two light waves and attains its maximum at the so-called Brillouin frequency ν_B . As the Brillouin frequency shift changes linearly with temperature and strain, a distributed temperature-strain sensor can be realized using SBS.

The theoretical model used to simulate the sensing in single-mode fiber is based on the wave equations for the pump and Stokes waves with the slowly-varying field amplitudes $A_p(z, t)$ and $A_s(z, t)$, respectively, which interact nonlinearly by the excitation of the acoustic wave $\rho(z, t)$ [4]:

$$\frac{n}{c} \frac{\partial A_p}{\partial t} + \frac{\partial A_p}{\partial z} = i \frac{\gamma_e \omega}{2nc\rho_0} \rho A_s \quad (1a)$$

$$\frac{n}{c} \frac{\partial A_s}{\partial t} - \frac{\partial A_s}{\partial z} = i \frac{\gamma_e \omega}{2nc\rho_0} \rho^* A_p \quad (1b)$$

where $A_p(z, t)$ and $A_s(z, t)$ [W] stand for the amplitudes of pump $E_p(z, t)$ and stokes $E_s(z, t)$ waves respectively, and n is the refractive index of the medium, c is the speed of light [m/sec], γ_e is the electrostrictive constant, ρ_0 is the unperturbed density [g/m³] and Γ_B is the Brillouin linewidth

[MHz] [4]. A^* represents the complex conjugate of the respective field. In the case of a transient SBS regime for nanosecond optical pulses participating in SBS, the slowly-varying amplitude approximation normally used may no longer be valid for the acoustic field. Therefore, the evolution of acoustic wave equation for the field $\rho(z, t)$ with second-order time derivative follows [5]:

$$\frac{\partial^2 \rho}{\partial t^2} + (\Gamma_B - i2\omega_B) \frac{\partial \rho}{\partial t} - i\Gamma_B \omega_B \rho = \frac{\gamma_e q_B^2}{4\pi} A_p A_s^* \quad (1c)$$

Simulation has proven to be a valuable tool in understanding SBS based distributed sensor systems. Accurate simulation models allow for the rapid design and optimization of both sensor infrastructure and signal processing techniques. Visualization of the spatiotemporal behavior of all three fields involved in SBS, at all points through the fiber, and is especially desirable in investigating phenomena whose origins may not be clear from the 'observable' data in an experiment [6]– [8].

If long pulses are assumed, the time dependence in Eq. (1) can be neglected, resulting in a simple and accurate solution to the problem. However, in the short pulse regime, the temporal derivatives cannot be neglected and the solution becomes more complex [9]. For this solution, high speed and accuracy with the low complexity is highly desirable. In general, for the solution of Eq. (1), conventional first order finite difference time domain (FDTD) solution schemes are preferred since they offer simplicity with respect to the form of the solution, although they suffer from several drawbacks, e.g. they require more spatial nodes to achieve the same accuracy as

higher order schemes, and this often results in longer computer run-times.

To solve Eq. (1), a numerical method based on the Simpson's rule to approximate temporal integrals was introduced by [10]. However, they used an implicit method and employed linearization by assuming that $|E_{sn}^{m+1}| \approx |E_{sn}^m|$ and this replacement weakens the coupling of the two laser fields and hence the results deviate from the exact solutions. In Ref. [6] the authors proposed a modified solution which is valid only when the time step size is equal to spatial step size. Moreover, their solution differs considerably from the real solution at locations where abrupt change takes place. Recently, we introduced the MOL solution of the SBS equations with appropriate flux limiter and sparse matrix integrator [11]. In this paper, we extend the results in Ref. [11] while computing a system of three second-order hyperbolic PDEs (Eqs. (1)).

In this paper we introduce, for the first time to the best of our knowledge, the solution of *transient SBS equations for nanosecond Stokes pulses* using Matlab ODE solvers [12]. By using this solution, the spectral and temporal evolution of distributed sensing based on stimulated Brillouin scattering (SBS) in optical fibers for several-nanosecond Stokes pulses is demonstrated. We introduce the solution technique and have adapted the software to provide a fast and stable solution by employing sparse matrix techniques and by using an appropriate flux limiter. Briefly, our solution technique combined with the Matlab software abilities provides a powerful method of simulating this SBS problem.

In arriving at a solution, the problem domain is first discretized over a finite grid and organized in a way that provides solutions for the optical and acoustic fields. The resolution of the solution can thus be varied to suit computational requirements. We have also employed an appropriate flux limiter to combat numerical damping/dispersion which avoids spurious oscillations in the solution. We have used software which identifies the sparsity pattern of the solution which is then supplied as an input to the solver. Additionally, we have determined the appropriate Matlab ode solver for this particular application. By use of an appropriate sparse matrix ode solver along with an effective flux limiter, for nanosecond pulse widths, it is shown that the computation can be successfully performed in a reasonable computation time without the damping factor exceeding 3%. Examples are presented, showing the utility of this efficient simulation technique.

2 PROBLEM FORMULATION

2.1 The Dynamics of Stimulated Brillouin Scattering (SBS)

In this paper we consider a system whereby continuous wave (CW) light, at frequency, ν_p , is injected into the fiber at position $z = L$. Pulses of light, known as Stokes pulses, are injected into the fiber at $z = 0$, at frequency, ν_s . Due to the SBS, there will be coupling of a counter propagating CW pump wave and a Stokes pulse wave via an induced acoustic wave. An en-

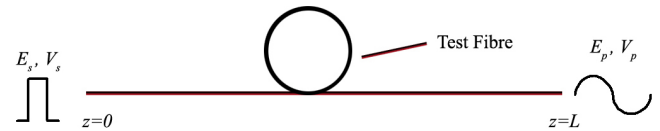


FIG. 1 Diagram of a typical BOTDA distributed sensor system. The energy is transferred from the CW pump to the low frequency Stokes pulse. The amount of loss of the pump is recorded at $z = 0$ as a function of the frequency difference in the form of the Brillouin spectrum, Thus the power of the CW field intensity over time at $z = 0$ corresponds to the available time domain information in a real sensing experiment.

hanced interaction between the two beams occurs when the frequency difference of the lasers matches the frequency of the longitudinal acoustic phonons of the optical fiber. There is then a transfer of energy from the high frequency beam (pump) to the low frequency beam (Stokes). The amount of loss of the pump is recorded at $z = 0$ as a function of the frequency difference in the form of a Brillouin spectrum and thus the power of the CW field intensity over time at $z = 0$ corresponds to the available time domain information in a real sensing experiment. The maximum loss occurs when the frequency difference of two beams matches the Brillouin frequency of the fiber. This maximum loss depends on the resonance detuning parameter, given by

$$\delta(z) = 2 * \pi \left[\nu_p - \nu_s - \delta\nu^{Res}(z) \right] \quad (2)$$

where $\delta\nu^{Res}(z)$ is the Brillouin frequency along the fiber and depends on the local strain and temperature. The resonance detuning parameter, $\delta(z)$, is the value of the pump-Stokes frequency shift for a resonant interaction at a given point. The resonance detuning parameter, $\delta(z)$ is dependent on the temperature and strain along the fiber. This dependence can form the basis of distributed fiber optic strain and temperature sensors.

A distributed Brillouin sensor, based on a Brillouin Optical Time Domain Analysis (BOTDA), is a device where Stokes and pump beams counter-propagate and where frequency and time dependent variations of the Stokes (or pump) intensity is detected. These sensors have been shown to provide the best performances in terms of sensing length, spatial resolution, temperature, and, strain accuracies [2, 13]. The interaction is a maximum when the optical frequency difference between pump (ν_p) and Stokes (ν_s) corresponds to the Brillouin frequency (ν_B). At the Brillouin frequency, the Stokes beam is amplified at the expense of the pump. ν_B is the longitudinal acoustic phonon frequency and is a local material signature. An increase (decrease) in temperature or strain induces a proportional increase (decrease) of ν_B . The sensor detects the Brillouin frequency variation as a function of position. The distributed nature of the sensor is achieved by field modulation of the probe beam. The topology of this sensor is shown in Figure 1 [14].

2.2 Theoretical Model

By making suitable transformation, Eqs. (1) can be written as in Eqs. (3) which is a system of three second-order hyperbolic

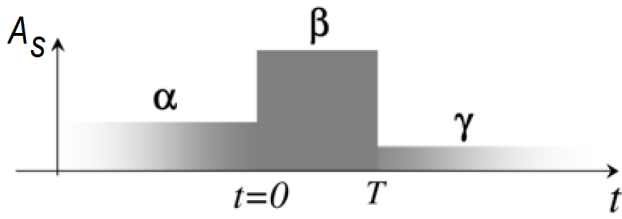


FIG. 1a Stokes coding for different waveforms. (a) Intensity pulse ($\alpha = \gamma = 0$ and $\beta > 0$); (b) Bright pulse with a CW component ($\alpha = \gamma > 0$ and $\beta > \gamma$); (c) dark pulse ($\alpha = \gamma > 0$ and $\beta = 0$); and (d) π -phase pulse ($\alpha = \gamma > 0$ and $\beta = -\gamma$). The pump pulse T determines the spatial resolution [15].

PDEs,

$$\partial E_p \partial t - \nu \frac{\partial E_p}{\partial z} = -E_a \cdot E_s \quad (3a)$$

$$\partial E_s \partial t + \nu \frac{\partial E_s}{\partial z} = E_a^* E_p \quad (3b)$$

$$\partial^2 E_a \partial t^2 + 2(\Gamma - j\Omega) \cdot \frac{\partial E_a}{\partial t} + \dots \quad (3c)$$

$$\left(\Omega_B^2 - \Omega^2 - 2j\Gamma\Omega \right) E_a = jgE_p E_s^*$$

where E_p , E_s , and E_a are the pump, Stokes, and acoustic fields, respectively, and j represents imaginary number $\sqrt{-1}$. $\Gamma = 1/\tau_{ph}$ is the relaxation rate $\omega = \omega_p - \omega_s$, ω_p , ω_s are the acoustic field, pump, and Stokes frequencies, respectively, Ω_B is the Brillouin frequency, $g = \nu\Gamma\Omega g_B$ and g_B is the SBS gain factor. Here, τ_{ph} is the phonon lifetime and it is 10ns for silica fibers.

2.3 Boundary and Initial Conditions

The CW light from a laser source representing the pump field, $E_p(z = L)$ is injected into the fiber at $z = L$. Therefore, the boundary condition of the pump wave is:

$$E_p(z = L, t) = E_p(z = L) \quad (4)$$

where $E_p(z = L)$ is the power of the CW laser. At the opposite end, the Stokes field, E_s , is introduced via a pulsed source. We assume a three-section pump pulse of the form (Fig. 1a):

$$E_s(z, t) = E_s^0 [\alpha \cdot u(t - z/v) + \dots \quad (5a)$$

$$(\beta - \alpha) \cdot u(t - t_0 - z/v) + \dots$$

$$(\gamma - \beta) \cdot u(t - t_0 - T - z/v)]$$

$$E_s(z = 0, t) = E_s^0 [\alpha \cdot u(t) + \dots \quad (5b)$$

$$(\beta - \alpha) \cdot u(t - t_0) + \dots$$

$$(\gamma - \beta) \cdot u(t - t_0 - T)]$$

where α, β, γ are amplitudes of the corresponding time intervals. Here $u(\cdot)$ represent the Heaviside unit step function. The first section of the pulse of height αA_s^0 and width t_0 enters the fiber ($z = 0$) at $t = 0$, followed by a βA_s^0 -high, T-wide middle section, ending with an infinitely long section of height γA_s^0 . In Matlab $u(\cdot)$ can be implemented via the function *heaviside* function. For the Bright pulse with a CW component waveform the base level of the Stokes pulse is characterized by the extinction ratio $ER = 10 \log(\beta/\alpha)$.

After propagating through the fiber, it is assumed that the pump and Stokes pulses exit the fiber without reflection. The

acoustic field is assumed to be zero everywhere at $t = 0$, and the CW light and Stokes leakage intensities turn on everywhere at $t = 0$. Therefore the initial conditions are [6],

$$E_p(z, t = 0) = E_p(z = L) \quad (6a)$$

$E_s(z, t = 0)$ is β for intensity pulse and α for the bright pulse with a CW component, dark pulse waveform and π -phase pulse, respectively.

$$E_a(z, t = 0) = 0 \quad (6b)$$

2.4 AC Detection of the pump output in the case of a finite CW component (base) of the Stokes pulse

Distributed sensing based on stimulated Brillouin scattering (SBS) is obtained by temporally resolved Brillouin spectra, i.e. the depleted output pump as a function of the detuning from the Brillouin frequency. The Brillouin frequency gives the strain and/or temperature along the fiber. The time-dependent output pump power or the time-dependent deviation from the stationary output power, induced by the Stokes pulse, is measured in the experiment. There are two possibilities of pump loss definition as DC or AC detection, respectively. However, AC detection is most likely to be used in the experiment to measure the small relative variations of the output pump signal in the case of several-nanosecond Stokes pulses. The pump loss defined as AC detection, $\alpha_{AC} = E_p(z = 0, t = 0) - E_p(z = 0, t)$, each as a function of the time and frequency detuning $\Delta\Omega = \Omega - \Omega_B$, represent the Brillouin spectrum. Here $E_p(z = 0, t)$ is the time-dependent output pump power at $z = 0$, $E_p(z = 0, t = 0)$ is the output pump power at the initial time moment, defined by the stationary pump-base SBS before the Stokes pulse arrives in the fiber [7].

3 NUMERICAL SOLUTION METHOD

3.1 Finite Difference Time Domain (FDTD) Solution

To obtain a solution to Eqs. (3a)-(3c), we follow the authors in Ref. [9] and substitute the following:

$$E_p = A_p e^{j\phi_p} \quad (7a)$$

$$E_s = A_s e^{j\phi_s}$$

$$E_a = A_a e^{j\phi_a} \quad (7b)$$

$$\frac{\partial E_a}{\partial t} = e^{j\phi_a} \left(\frac{\partial A_a}{\partial t} + jA_a \left(\frac{\partial \phi_a}{\partial t} \right) \right)$$

$$\frac{\partial^2 E_a}{\partial t^2} = e^{j\phi_a} \left\{ \frac{\partial^2 A_a}{\partial t^2} - A_a \left[\left(\frac{\partial \phi_a}{\partial t} \right)^2 \right] \right\} + \dots \quad (7c)$$

$$j e^{j\phi_a} \left\{ 2 \frac{\partial A_a}{\partial t} \frac{\partial \phi_a}{\partial t} + A_a \left(\frac{\partial^2 \phi_a}{\partial t^2} \right) \right\}$$

into Eqs. (3a)-(3c) and with making transformation $z' = \frac{z}{1/v} \cdot \gamma_a$ to remove v yields the equations,

$$\frac{\partial A_s}{\partial t} + \frac{\partial A_s}{\partial z} = A_p A_a \cdot \cos(\Phi) \quad (8a)$$

$$\frac{\partial A_p}{\partial t} - \frac{\partial A_p}{\partial z} = -A_s A_a \cdot \cos(\Phi) \quad (8b)$$

$$\frac{\partial^2 A_a}{\partial t^2} = -g A_p A_s \cos \Phi - \dots$$

$$\left\{ 2\Gamma \frac{\partial A_a}{\partial t} + 2A_a \Omega \left(\frac{\partial A_a}{\partial t} \right) \right\} - \dots \quad (8c)$$

$$(\Omega_B^2 - \Omega^2) A_a + A_a \left[\left(\frac{\partial A_a}{\partial t} \right) \right]^2$$

$$\frac{\partial \phi_s}{\partial t} + \frac{\partial \phi_s}{\partial z} = -\frac{A_p A_a}{A_s} \cdot \sin(\Phi) \quad (8d)$$

$$\frac{\partial \phi_p}{\partial t} - \frac{\partial \phi_p}{\partial z} = -\frac{A_s A_a}{A_p} \cdot \sin(\Phi) \quad (8e)$$

$$A_a \left(\frac{\partial^2 \phi_p}{\partial t^2} \right) = -g A_p A_s \cos \Phi - \dots$$

$$2 \left(\frac{\partial \phi_a}{\partial t} \right) \left\{ \frac{\partial A_a}{\partial t} + \Gamma A_a \right\} + \dots \quad (8f)$$

$$2\Omega \left\{ \left(\frac{\partial A_a}{\partial t} \right) + \Gamma A_a \right\}$$

Where, $\Phi = \Phi_a + \Phi_s - \Phi_p$. The steady state is obtained by setting the time derivatives to zero. Combining Eqs. (8c) and (8f) yields $\tan \Phi = -(\Omega_s^2 - \Omega^2)/(2\Gamma\Omega)$. Taking the positive part of the amplitudes, from Eq. (8c) it can be concluded that,

$$\cos \Phi = \frac{2\Gamma\Omega}{\sqrt{4\Gamma^2\Omega^2 + (\Omega_B^2 - \Omega^2)^2}} \quad (8g)$$

$$\sin \Phi = -\frac{\Omega_s^2 - \Omega^2}{\sqrt{4\Gamma^2\Omega^2 + (\Omega_B^2 - \Omega^2)^2}}$$

The basic idea of the method of lines (MOL) is to replace the spatial derivatives with algebraic approximations [16]. Since, spatial variables are discretized and time is used as the continuous variable, this approach is called a *semidiscretization*. This effectively removes the spatial derivatives from the PDE and, since only the initial value independent variable remains, e.g., t , the PDE has been converted to a system of approximating ordinary differential equations (ODEs) that can be integrated by standard, well-established numerical algorithms for initial value ODEs.

Eqs. (3a)-(3c) are formulated by making some transformations which are $t_{real} \cdot \gamma_a = t_{transformed}$, $z_{real} \cdot \gamma_a/v = z_{transformed}$, where v is the speed of light in the fiber and γ_a is $110 \cdot 10^6$. Thus, the problem is discretized on a domain of,

$$(Z_{transformed}, t_{transformed}) \in \left\{ \begin{array}{l} 0 \leq \frac{z_{real}}{v} \gamma_a \leq \text{fiber length} \\ 0 \leq t_{real} \cdot \gamma_a \leq \frac{\text{fiber length}}{v} \end{array} \right\} \quad (9)$$

The solution is obtained on uniform grids in z and t ,

$$z_{transformed,i}, t_{transformed,j} :=$$

$$(z_i, t_j) : \left\{ \begin{array}{l} z_{transformed,i} = i \cdot \Delta z \quad 0 \leq i \leq N \\ t_{transformed,j} = j \cdot \Delta t \quad 0 \leq j \leq M \end{array} \right\}$$

For the MOL solution of Eq.(1b), $E_p(z, t)$ can be approximated by using the upwind finite difference (FD) approximations

for the first derivatives in z so that $E_p \approx (E_{p_i} - E_{p_{i-1}})/\Delta z$, where i is an index designating a position along a grid in z and Δz is the spacing in z along the grid ($i - 1$ designates the upwind direction). For the MOL solution of Eq.(1b), $E_s(z, t)$ can also be approximated by upwind FD approximations, i.e., $E_s \approx (E_{s_{i+1}} - E_{s_i})/\Delta z$ ($i + 1$ designates the upwind direction).

For the discretization of Eqs. (8a)-(8f), variables are defined as $y1 = A_s$, $y2 = A_p$, $y3 = A_a$, $y4 = dy3/dt$, $y5 = \Phi_a$, $y6 = dy5/dt$. The boundary conditions are $y1(z = 0, t) = E_s(z = 0, t)$, $y2(z = L, t) = E_p(z = L)$. The initial conditions are, $y1(z, t = 0) = \beta$ for intensity pulse and α for the bright pulse with a CW component, dark pulse waveform and π -phase pulse, respectively, $y2(z, t = 0) = E_p(z = L)$, $y3(z, t = 0) = 1 \cdot 10^{-11}$, $y4(z, t = 0) = 0$, $y5(z, t = 0) = 0$, $y6(z, t = 0) = 0$. By discretizing Eqs. (10a)-(10f) in the space variable z , the resulting system of ODEs is as follows:

$$\frac{dy1_i}{dt} = -(y1_i - y1_{i-1})/(x_i - x_{i-1}) - \dots \quad (10a)$$

$$y2_i \cdot y3_i \cdot \cos(\Phi) \quad 2 \leq i \leq N$$

$$y1_1 = E_s(z = 0, t)$$

$$\frac{dy2_i}{dt} = -(y2_i - y2_{i+1})/(x_i - x_{i+1}) + \dots \quad (10b)$$

$$y1_i \cdot y3_i \cdot \cos(\Phi) \quad 2 \leq i \leq N$$

$$y2_N = E_p(z = L)$$

$$\frac{dy3_i}{dt} = y4_i \quad (10c)$$

$$\frac{dy4_i}{dt} = -g \cdot y1_i \cdot y2_i \cdot \sin(\Phi) - \dots \quad (10d)$$

$$(2 \cdot \Gamma \cdot y4_i + 2 \cdot y3_i \cdot \Omega \cdot y6_i) + \dots$$

$$y3_i \cdot (y6_i)^2 - y3_i \cdot (\Omega_b^2 - \Omega) \quad 1 \leq i \leq N$$

$$\frac{dy5_i}{dt} = y6_i \quad 1 \leq i \leq N \quad (10e)$$

$$\frac{dy6_i}{dt} = (-g \cdot y1_i \cdot y2_i \cdot \sin(\Phi) + \dots \quad (10f)$$

$$2 \cdot \Omega \cdot (y4_i + \Gamma \cdot y3_i) - \dots$$

$$2 \cdot y6_i \cdot (y4_i + \Gamma \cdot y3_i) / y3_i \quad 1 \leq i \leq N$$

The handling of Eqs.(10a)-(10f) is discussed in the appendix. The solver integrates the PDEs via the Matlab statement, `[t,u]=ode23t(@pde,tout,u0,options);` The initial conditions $u0$, transformed time interval $tout$, and options, which include tolerances and Jacobian sparsity information, are supplied to the solver.

3.2 Application of a Sparse Matrix Integrator

For a large system of ODEs, it is typical that only a few components of y appear in each equation. If component j of y does not appear in component i of $f(t, y)$, then the partial derivative $\partial f_i / \partial y_j$ is zero. If most of the entries of a matrix are zero, the matrix is said to be *sparse*. By storing only the nonzero entries of a sparse Jacobian, storage is reduced from the square of the number of equations d to a modest multiple of d . If the Jacobian is sparse then so is the iteration matrix. As with storage, the cost of solving linear systems by elimination can be

reduced dramatically by paying attention to zero entries in the matrix. By taking into account the known value of zero for most of the entries in the Jacobian, it is typically possible to approximate all the nonzero entries of several columns at a time. An important special case of a sparse Jacobian is one that has all its nonzero entries located in a band of diagonals [17, 18].

If the system of PDEs could be represented purely by *banded Jacobians*, then a banded integrator would be even more efficient than the *sparse integrator* since the banded integrator would know in advance where the nonzero elements occur (they all are in the band). It would therefore not have to search for the nonzero elements, and follow the resulting logic to use these nonzero elements, all of which add complexity to a sparse integrator. However, these nonzero elements are located/displayed with the Matlab command `spy()`. For the PDE systems of Eqs. (10a)-(10f), some of the nonzero elements are located along the main diagonal as well as the others located outside the band. Therefore, for this case, a *sparse integrator* is particularly effective since it also operates on out-of-band elements, the so-called *outliers*, in performing the numerical integration. Thus, this combination of elements in a band along the main diagonal, plus outliers, leads to the efficiency of sparse matrix integration [11].

Since this particular application has a main diagonal, plus outliers, we have used the routine `jpattern_num()` to produce a Jacobian map for the sparse matrix option of the *ode* solver. In this routine, the elements of the Jacobian matrix are computed by finite differences (FDs). This requires a base point, or base points, around which the numerical derivatives are computed. A base point can be any value within the range of the variation of the associated dependent variable. A precise value is not required (but if the *ode* solver fails, some experimentation with this value may be required). In our application we set the base points `ybase(i)`, equal to the initial condition for each dependent variable which is sufficiently accurate to provide a good solution.

In order to calculate the partial derivatives in the Jacobian matrix by FD approximations, the derivatives at the base point, dy_i/dt , are also required. Note that the routine `pde()` (see Appendix), that calculates the derivatives dy_i/dt in Eqs. (10a)-(10f) is called by the Matlab command:

```
ytbase=pde(tbase,ybase);
```

The elements of the Jacobian matrix are evaluated numerically by the Matlab command:

```
[Jac,fac]=numjac(@pde,tbase,ybase,ytbase,thresh,fac, vectorized);
```

where the following argument values are used: `tbase=0`, `thresh=1.10-16` and `fac=[]`. The input variable `thresh` represents the *threshold of significance* for $Y(i)$, such that $Y(i) < \text{thresh}$ is not important; the input variable `fac` represents a working storage column vector and should not be changed between calls; and `vectorized = 'off'` or `'on'` tells `numjac()` whether *single* or *multiple* values of Y respectively can be obtained with a single function evaluation.

Once the Jacobian matrix has been defined by this call to `numjac()`, a map of the Jacobian is plotted by the code that follows the call to `numjac()`. With a call to the routine `spones()`,

Jacobian sparsity pattern – nonzeros 5981 (0.185%)

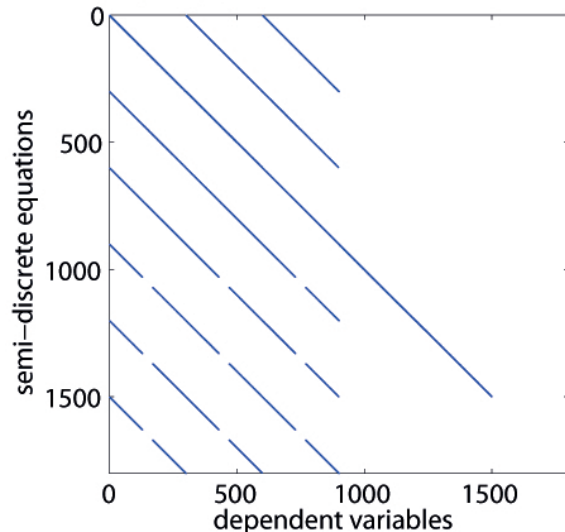


FIG. 2 Sparsity Pattern of the Eqs.(10a)-(10f). The solution is obtained with $n = 300$

each *non-zero element* is replaced by a "1" so as to create a "0-1" map of the Jacobian matrix [11, 16].

```
S=spones(sparse(Jac));
```

In Figure 2, the sparsity pattern of Eqs. (10a)-(10f) is plotted by a subsequent call to the routine `spy(S)`. Code for `jpattern_num()` is available as a library routine which can be downloaded from Ref. [19].

Note that only 0.185 % of the $6 \times 300 \times 6 \times 300$ elements of the Jacobian map of Figure 2 are nonzero. This is not uncommon; that is, most of the elements are zero. This condition is the reason for the use of sparse matrix integrators since they will in general process only the *nonzero elements* and will not expend computer time processing the many zero elements. The processing time with and without `jpattern_num()` is illustrated in Table 1 for the implicit (stiff) ODE solvers (`ode15s`, `ode23s`, `ode23t`, and `ode23tb`). The computation is performed on a i5 Core, 4 GB RAM, Laptop computer running the Windows 7 operating system. From Table 1, it can be seen that the sav-

Run Time (sec)		
AbsTol and RelTol= 1×10^{-5} , Pulse width=5 ns.		
Without Flux-limiter		
n=300	Without Jpattern_num	With Jpattern_num
ode15s	204	58
ode23t	242	70
ode23tb	292	84
ode23s	Computation failed	122

Table 1: Run-time comparison of implicit ODE solvers with/without use of `jpattern_num()`

ing in computer time can be very substantial. It is clear that the additional logic of the sparse matrix integrator (to detect the nonzero elements of the Jacobian matrix and then perform the numerical integration using only these nonzero elements) is well worth the additional complexity of the coding for the sparse matrix. Moreover, for `ode23s`, using `Jpattern_num()` is the difference between success and failure.

3.3 Use of a Flux Limiter

Numerical dissipation is generally observed when solving strongly convective or strongly hyperbolic PDEs. The main cause of the dissipation is low order approximation of the discretization. It should be noted that numerical dissipation produced by first order approximations is to be expected due to truncation of the Taylor series that is the basis for the approximations. In recent years, various high resolution schemes have been developed to obviate this effect with a high degree of accuracy, albeit at the expense of algorithmic and computational complexity. Examples of particularly effective schemes are based upon flux/slope limiters [20] and WENO methods [21].

Flux limiters are used in numerical schemes to solve problems in science and engineering that are described by partial differential equations (PDEs). Their main purpose is to avoid the spurious oscillations (wiggles) that would otherwise occur with high-order spatial discretization due to shocks, discontinuities, or steep gradients in the solution domain. They can be used directly on finite difference schemes for simple applications. Use of flux limiters, together with an appropriate high-resolution scheme, makes the solutions *total variation diminishing*. Flux limiters are also referred to as *slope limiters* because they both have the same mathematical form and both have the effect of limiting the solution gradient near shocks or discontinuities. In general, the term *flux limiter* is used when the limiter acts on system *fluxes*, and *slope limiter* is used when the limiter acts on system *states*. The main idea behind the construction of flux limiter schemes is to limit the spatial derivatives to realistic values; this usually means physically realizable values. They are used in high-resolution schemes for solving problems described by PDE's and only come into operation when sharp wave fronts are present [22].

Considering the *semidiscrete* scheme below,

$$\frac{du_i}{dt} + \frac{1}{\Delta x_i} [F(u_{i+\frac{1}{2}}) - F(u_{i-\frac{1}{2}})] = 0 \tag{11a}$$

where, for a *finite difference scheme*, $F(u_{i+1/2})$ and $F(u_{i-1/2})$ represent flux values on the grid at point $x = x_{i+1/2}$ and point $x = x_{i-1/2}$. If these fluxes can be represented by *low-* and *high* resolution schemes, then a flux limiter can switch between these schemes depending upon the gradients close to the particular cell as follows:

$$F(u_{i+\frac{1}{2}}) = f_{i+\frac{1}{2}}^{low} - \phi(r_i)(f_{i+\frac{1}{2}}^{low} - f_{i+\frac{1}{2}}^{high}) \tag{11b}$$

$$F(u_{i-\frac{1}{2}}) = f_{i-\frac{1}{2}}^{low} - \phi(r_{i-1})(f_{i-\frac{1}{2}}^{low} - f_{i-\frac{1}{2}}^{high}) \tag{11c}$$

where, f^{low} low-resolution flux, f^{high} high-resolution flux, $\phi(r)$ = flux limiter function, and r represents the ratio of successive gradients on the solution mesh, i.e.,

$$r_i = (u_i - u_{i-1}) / (u_{i+1} - u_i) \tag{11d}$$

The limiter function is constrained to be greater than or equal to zero, i.e., $r \geq 0$. Therefore, when the limiter is equal to zero (sharp gradient, opposite slopes, or zero gradient), the flux is represented by a *low-resolution scheme*. Similarly, when

the limiter is equal to 1 (smooth solution), it is represented by a *high-resolution scheme*. The various limiters listed below have differing switching characteristics and are selected to suit the particular problem and numerical solution scheme. No particular limiter has been found to work well for all problems, and a particular choice is usually made on a trial-and-error basis [11,22].

$$\text{van Leer: } \phi_{vl} = \frac{r + |r|}{1 + r}; \quad \lim_{r \rightarrow \infty} \phi_{vl}(r) = 2 \tag{12a}$$

$$\text{smart: } \phi_{sm}(r) = \max [0, \min(2r, (0.25 + 0.75r), 4)]; \quad \lim_{r \rightarrow \infty} \phi_{sm}(r) = 4 \tag{12b}$$

$$\text{superbee: } \phi_{sb}(r) = \max [0, \min(2r, 1, \min(r, 2))]; \quad \lim_{r \rightarrow \infty} \phi_{sb}(r) = 2 \tag{12c}$$

The parameters of the flux limiter, which available as a library routine which can be downloaded from [23], are as follows,

```
y1x=-flux'function(xl,xu,n,y1,1);
y2x= flux'function(xl,xu,n,y2,-1);
```

The fifth parameter is direction of the flow for the linear advection equation $u_t + cu_x = 0$. This argument is required, since the limiter requires the direction of flow or wave propagation. It takes the values +1 or -1 depending on the direction of the upwinding of the FD. The fifth parameter value of -1 for y2x indicates that the pulse moves right to left in z at velocity $c = -1$ (i.e., counter propagating in z). The spatial domain in z is defined as $xl \leq z \leq xu$ with n points. The interval $xu - xl$ is determined by the fiber length.

It should be noted that when applying flux limiter solutions it is necessary to have sufficient grid points (say 10 or more) between the occurrence of adjacent steep gradients. If insufficient grid points are used, then smearing of the solution may occur or, in extreme cases, the solution could blow up.

In order to establish the best flux limiter function to suit our particular application we performed numerical experiments to compute the degree of smearing. In Figure 3, The effect of *van Leer*, *Smart* and *Suberbee* limiters in terms of smearing can be seen.

In Figure 3(a), it is clearly seen that the Stokes magnitude is damped due to the numerical dissipation. In Figure 3(b)-(e), the effect of different flux limiters on the Stokes pulse is demonstrated. Damping which can be considered as the amplitude ratio at $z = 0$ and at $z = L$ is calculated as, (b) 62.2 % for *van Leer*, (c) 55.44 % for *smart* (d) 47.74% for *superbee* flux limiter, respectively (n=100). In Figure 3(e), the damping effect is decreased to only 2.8 % with *superbee* flux limiter for 300 discretization points. In Figure 3(b)-(e), it is clearly seen that, for this particular application, the *superbee* limiter is more capable of avoiding numerical damping compared to the performance of *van Leer* and *Smart* flux limiters. Since it has superior performance for this application, the *superbee* flux limiter is preferred.

4 EXAMPLE

The simulation models an optical fiber, 15 m in length, and the parameter values of the simulation are inferred from the pa-

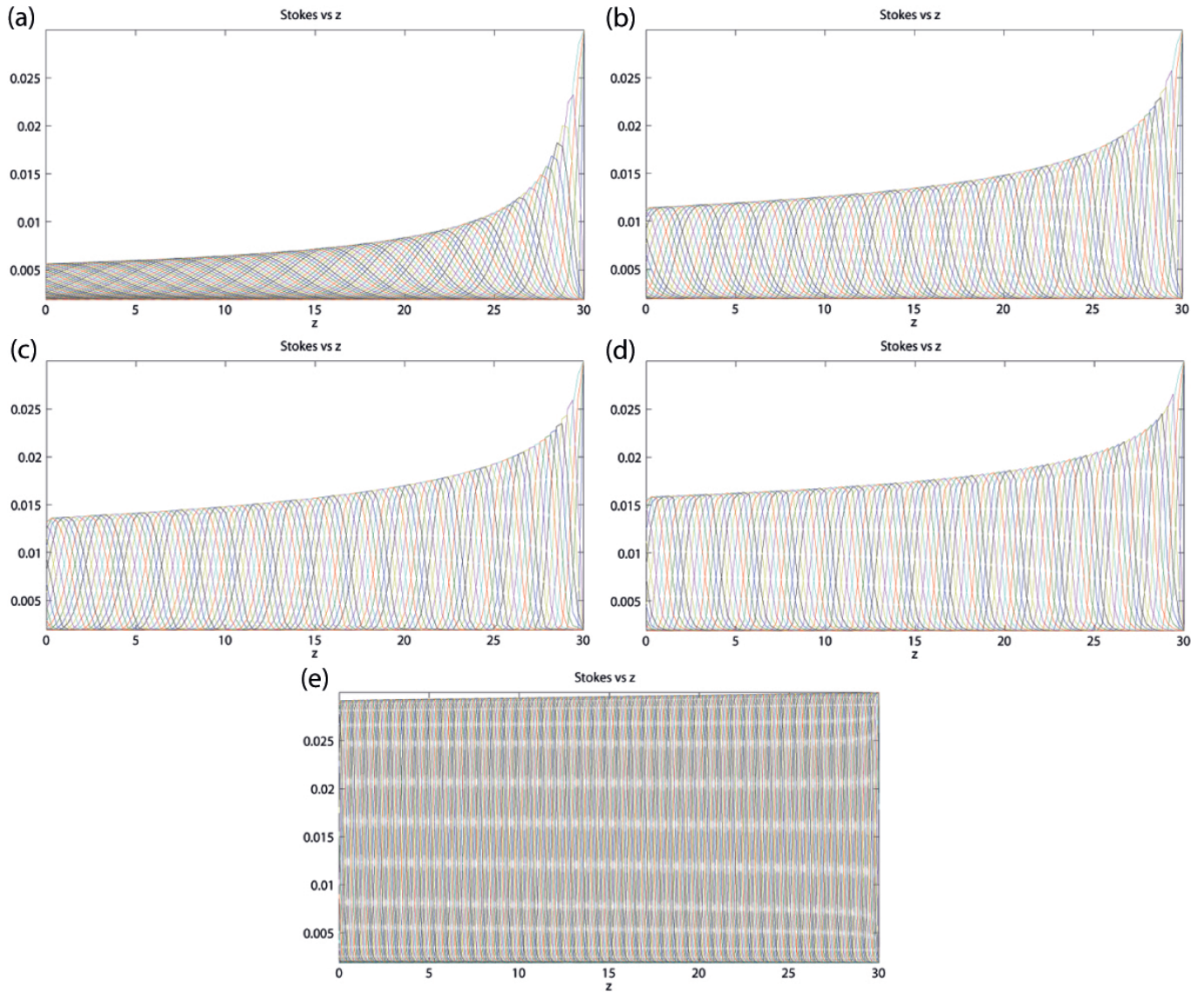


FIG. 3 The simulation of the pulse, (a) without using flux limiter, with using (b) van Leer, (c) smart (d) superbee flux limiter with $n=100$ discretization points. In (e) superbee flux limiter is used with $n=300$ discretization points.

per [7]. The Brillouin frequency, $\delta\nu^{Res}(z)$, is 12.800 GHz along the fiber, except for a specified section in the center where its $\delta\nu^{Res}(z)$ is different from the main fibers' Brillouin frequency, representing a strained section. The moment in time, when the Stokes pulse enters the fiber is assumed to be equal to t_0 with respect to the pulse leading edge. Bright pulse with a CW component is used as a Stokes waveform. The other relevant parameters are: CW power = 5 mW; Stokes power = 10 mW (β); Stokes pulse rise time = 100ps. The simulation models a pulse on-resonance, with the strained section ($\nu_p - \nu_s$ is varied). The other Brillouin parameters typical for the single-mode silica fibers at the wavelength 1.3 μm : $\tau_{ph} = 10\text{ns}$, $A_{eff} = 50 \mu\text{m}^2$, $g_B = 5 \times 10^{-11} \text{ m/W}$, $\nu = 0.2 \text{ m/ns}$.

The expected result of this simulation is that the Stokes pulse will propagate down the section of fiber with no loss or dispersion. Since the fiber's natural $\delta\nu^{Res}(z)$ is much different from $\nu_p - \nu_s$, there should be little interaction in the 'unstrained' portions of the fiber. Upon entering the 'strained' section, the pulse will interact with the CW field, causing a rise in the acoustic field intensity. The CW field will be depleted, and

the pulse will be enhanced. The range of the depletion depends on the extinction ratio and the interaction is stronger for a low extinction ratio. After the Stokes pulse passes through the strained section, the acoustic field will decay away and the CW field will return to its steady state value. The effect of the interaction will take the form of a depleted region of CW light propagating to the end of the fiber [6].

Figure 4 shows the Stokes, CW, and acoustic fields, simulated with and without the *superbee* flux limiter. As can be seen in Figure 4(a), dissipation and damping is clearly observed. Over time, the Stokes pulse broadens, as well as decreases in intensity. However, the expectation from this simulation is that Stokes pulse will propagate down the section of fiber with no loss or dispersion. It must be noted that, in the 'unstrained' portions of the fiber $\Delta \neq 0$ i.e., $\nu_p - \nu_s \neq \delta\nu^{Res}(z)$, in the "strained" section of the fiber $\Delta = 0$, i.e., $\nu_p - \nu_s = \delta\nu^{Res}(z)$. Plots (c) and (d) show the fields simulated with the *superbee* flux limiter. Especially in plot (c), with flux limiter, it can be clearly seen that damping is suppressed and the ratio of the amplitudes at $z = 0$ and at $z = L$ becomes 0.972 which means

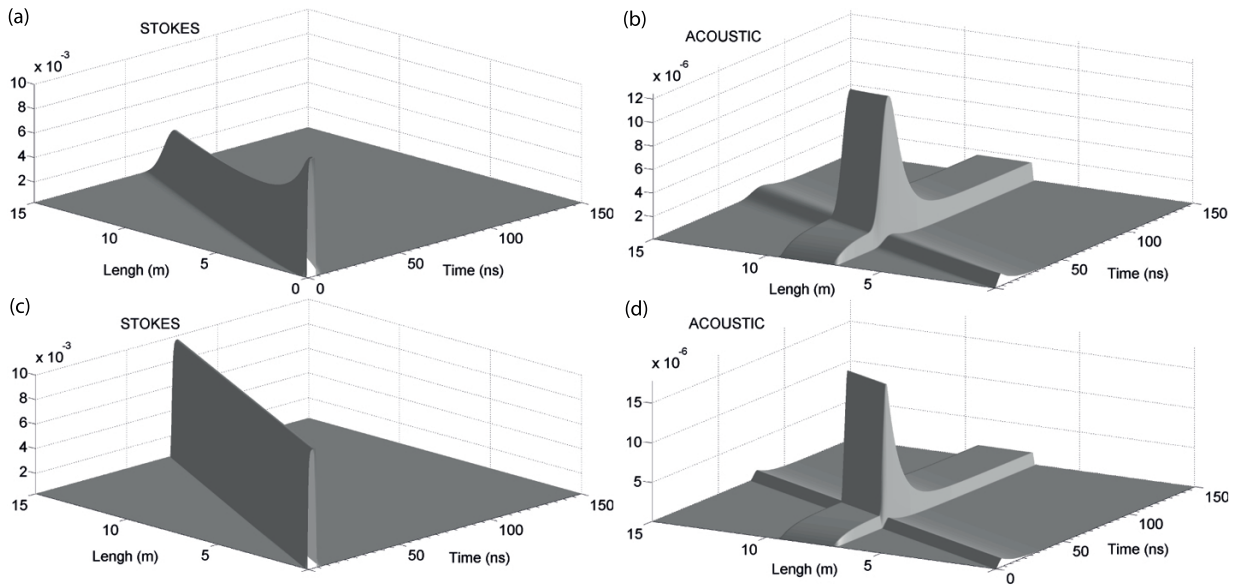


FIG. 4 Using the simulation of Eqs. (10a)-(10f) with $n = 190$ discretization points and $t_0 = 0$ ns, evolution of the Stokes (y_1), and Acoustic (y_3) fields for a 15 m section of fiber is plotted in Figure 4(a)-(c), (b)-(d), respectively. The frequency difference of Stokes pulse wave was set to be on resonance with a 2 m section in the center of the fiber. Since, in plots (a), (b), flux limiter is not used, the dispersion and attenuation is evident. Plots (c), (d), and show the results when the flux limiter is used. Compared with the low order discretization, with the introduction of the flux limiter, dispersive attenuation (smoothing of sharp edges) along the fiber obviously is eliminated.

that Stokes wave lost only 2.8% of its magnitude. Another expectation of the simulation is that, when the pulse enters the strained section there should be a rise in the acoustic field intensity. However, the *superbee* flux limiter prevents this rise during stabilization.

For the case of a 4-ns Stokes pulse with ER = 15 dB and 50 dB, in Figure 5, AC detection is plotted as a function of the time and frequency detuning $\Delta\Omega = \Omega - \Omega_B$. Here t_0 and fiber length is assumed to be 5 ns and 15m, respectively. As can be seen in Figure 5, AC detection shows different characteristics at the time $t < t_0$, before the probe pulse arrives in the fiber, and at $t > 2L/\nu + t_0 + \tau_s$ (159 ns), after it has left the fiber and the related pump depletion has propagated from $z = 0$ to the output $z = L$. The AC-detected Brillouin spectrum shows a Lorentzian line shape with the Brillouin natural width $\Delta\nu_B = \Gamma/\pi \approx 32\text{MHz}$ resulting from the pump-base SBS and this corresponds to no signal before pulse arrival ($t < t_0$) and transient relaxation again to zero signal through complicated Brillouin loss signal oscillations with even negative values when out-of-resonance ($t > 2L/\nu + t_0 + \tau_s$).

Now we address the SBS-based sensing for longitudinally inhomogeneous distribution of the Brillouin frequency along the fiber. First we consider AC-detected Brillouin spectra of the fiber consisting of two 10-m long sections with Brillouin frequencies 12.800 and 12.840 GHz for the case of a 4-ns Stokes pulse (Figure 6). We verify how the sections and especially the boundary between them are identified AC detection for two different base levels of 15 and 50 dB. The thick dashed curve is drawn at the time $t = 157$ ns, when the pulse peak center is passing through the middle of the fiber, and this point should be identified by sensing as a boundary between the sections. Additionally, at the moment in time corresponding to the boundary, the spectrum transition from one line to another is clearly visible. However, due to broad pulse spectrum this transition is complicated by the prolonged relaxation of

the acoustic field at the frequency related to the previous section, as the pulse has already progressed to the next section.

Figure 7 illustrates the SBS-based sensing of a 20-cm section in the middle of a 10-m fiber, where the Brillouin frequency of the section is shifted by 80 MHz relative to that of the rest of the fiber. In Figure 7, we have used a 2-ns Stokes pulse with ER = 15 dB (Figure 7(a)) and 50 dB (Figure 7(b)), respectively. The other parameters are taken from Figure 5. Thick dashed curves are drawn for the output pump at times when the pulse peak center passes through the section's center, and at the frequency shift corresponding to the Brillouin frequency of the section. They denote the time and the frequency, which should be revealed from the detection as a section's spatial and spectral "location".

However, it was observed that, for pulses substantially shorter than the acoustic lifetime in silica, the measured gain profiles abandon their expected spectral broadening and the linewidth was shown to gradually return to the natural value, as determined by the acoustic lifetime [15]. This was explained by the fact that in a real experiment the Stokes pulses have a small cw component (base) due to unavoidable leakage of any optical amplitude modulator used to generate nanosecond pulses. In this case the linewidth could still be close to the natural linewidth, provided the relative contribution of the transient regime on the output pump depletion is weak [7]. In our numerical experiment the same trend is observed in Figure 8.

5 CONCLUSIONS

In this paper we introduce, for the first time to the best of our knowledge, the solution of *transient SBS equations for nanosecond Stokes pulses* with different waveforms using Matlab ODE solvers (MATLAB, 2012). By using this solution the spectral and temporal evolution of the AC detection is

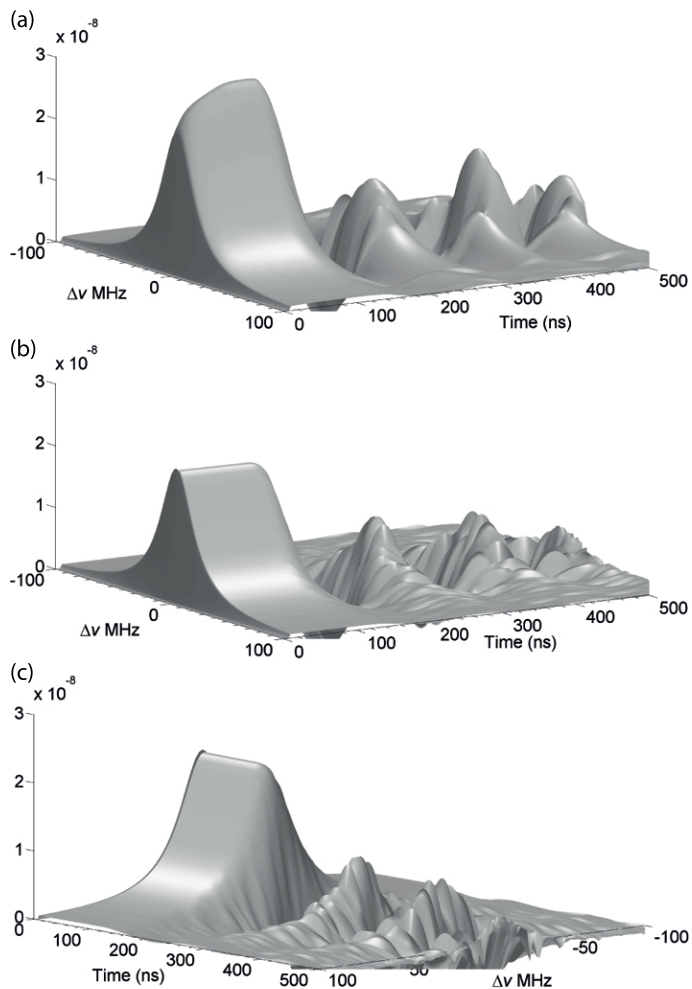


FIG. 5 Brillouin spectra resolved in the time domain by AC detection for the pump power $P_p = 5$ mW and the Stokes pulse with duration $\tau_s = 4$ ns, $t_0 = 5$ ns, peak power $P_s = 10$ mW, extinction ratio (a) ER = 15 dB (b) ER = 50 dB, (c) ER = 50 dB with back view of Brillouin spectra, for the fiber length $L = 15$ m and $n=190$ discretization points.

simulated. The MOL scheme provided the basis for a solution which was adapted to a suitable mesh size allowing rapid calculation of the fields involved in the SBS interaction. We introduce the solution technique and have adapted the software to provide a fast and stable solution by employing sparse matrix techniques and by using an appropriate flux limiter. The simulation method presented here can be used to foster a rapid understanding of the internal dynamics of the distributed sensing based on SBS, lending insight for both experimental design and interpretation of results.

6 ACKNOWLEDGEMENTS

The University of Hasan Kalyoncu Optical Fibre Systems Laboratory is supported in part by grants from the TUBITAK.

References

[1] T. Horiguchi, K. Shimizu, T. Kurashima, M. Tateda, and Y. Koyama, "Development of a distributed sensing technique using Brillouin scattering," *J. Lightwave Technol.* **13**, 1296–1302 (1995).

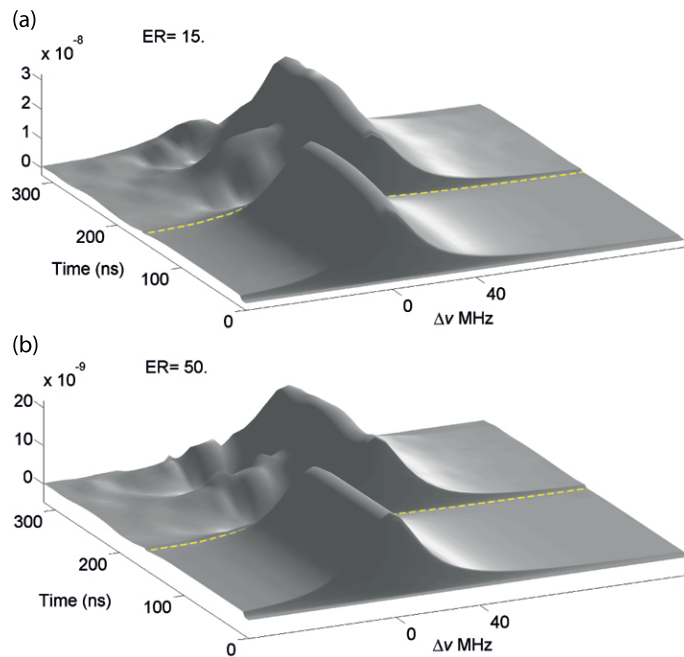


FIG. 6 SBS-based sensing of the boundary between two 10-m fibers with 40 MHz-shifted Brillouin frequencies by a 4-ns Stokes pulse with (a) 15-dB (b) and 50-dB (c) base for AC detection, and the same other parameters as in Figure 5. The thick dashed curve is at time moment $t = 157$ ns and corresponds to the boundary between the fiber sections. $n=195$ discretization points

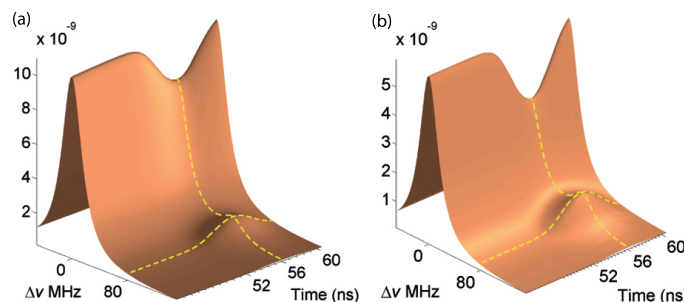


FIG. 7 SBS-based sensing of a 20-cm section with 80MHz-shifted Brillouin frequency in the middle of a 10-m fiber by a $\tau_s = 2$ -ns Stokes pulse with (a) 15-dB and (b) 50-dB base for AC detection and the same other parameters as in Figure 5. Thick dashed curves are at the time moment $t = 56$ ns and the frequency detuning $\Delta\nu = 80$ MHz. For each figures $n=400$ and Bright Stokes pulse with a CW component is used as a waveform.

[2] X. Bao, J. Dhlwayo, N. Heron, D. Webb, and D. Jackson, "Experimental and theoretical studies on a distributed temperature sensor based on Brillouin scattering," *J. Lightw. Technol.* **15**, No. 10, 1842–1851 (1997).

[3] X. Bao, M. DeMerchant, A. Brown, and T. Bremner, "Tensile and compressive strain measurement in the lab and field with the distributed Brillouin scattering sensor," *J. Lightwave Technol.* **19**, 1698–1704 (2001).

[4] R. Boyd, *Nonlinear Optics* (Academic Press, San Diego, 2003).

[5] I. Velchev, D. Neshev, W. Hogervorst, and W. Ubachs, "Pulse compression to the subphonon lifetime region by half-cycle gain in transient stimulated Brillouin scattering," *IEEE J. Quant. Electron.* **35**, 1812–1816 (1999).

[6] A. Marble, K. Brown, and B. Colpitts, "Stimulated Brillouin scattering modeled through a finite difference time domain approach," *Proc. SPIE* **5579**, 404–414 (2004).

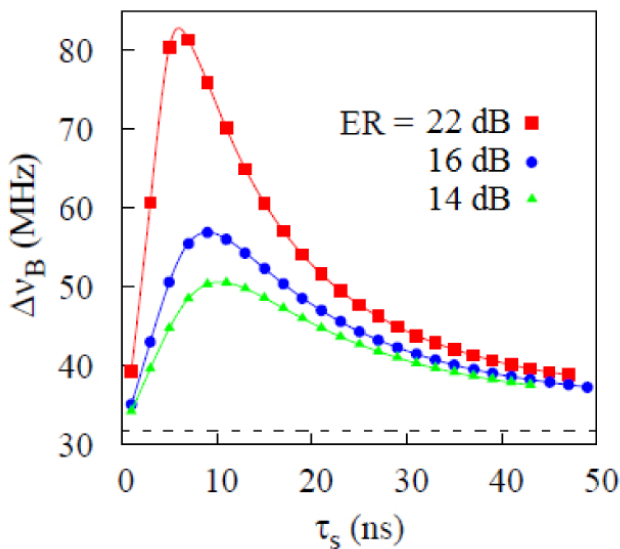


FIG. 8 FWHM versus Stokes pulse duration for AC detection at $t = 157$ ns, different extinction ratio and the same pump and Stokes pulse parameters as in Figure 5. Dashed line shows the natural Brillouin linewidth.

[7] V. Kalosha, E. Ponomarev, L. Chen, and X. Bao, "How to obtain high spectral resolution of SBS-based distributed sensing by using nanosecond pulses," *Opt. Express* **14**, 2071-2078 (2006).

[8] A. Minardo, R. Bernini, and L. Zeni, "Numerical analysis of single pulse and differential pulse-width pair BOTDA systems in the high spatial resolution regime," *Opt. Express* **19**, 19233-19244 (2011).

[9] C. Chow, and A. Bers, "Chaotic stimulated Brillouin scattering in a finite-length medium," *Phys. Rev. A* **47**, 5144-5150 (1993).

[10] R. Chu, M. Kanefsky, and J. Falk, "Numerical study of transient stimulated Brillouin scattering," *J. Appl. Phys.* **71**, 4653 (1992).

[11] F. Gokhan, G. Griffiths, and W. Schiesser, "Method of lines solutions for the 3-wave model of Brillouin equations," *Engineering Computations* (2013)

[12] http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf

[13] L. Zou, X. Bao, V. Afshar, and L. Chen, "Dependence of the Brillouin frequency shift on strain and temperature in a photonic crystal fiber," *Opt. Lett.* **29**, 1485 (2004).

[14] F. Ravet, L. Chen, X. Bao, L. Zou, and V. Kalosha, "Theoretical study of the effect of slow light on BOTDA spatial resolution," *Opt. Express* **14**, 10351-10358 (2006).

[15] J. Beugnot, M. Tur, S. Mafang, and L. Thévenaz, "Distributed Brillouin sensing with sub-meter spatial resolution: modeling and processing," *Opt. Express* **19**, 7381-7397 (2011).

[16] W. Schiesser, and G. Griffiths, *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab* (Cambridge University Press, New York, 2009).

[17] L. Shampine, and M. Reichelt, "The MATLAB ODE Suite," *SIAM J. Sci. Comput.* **18**, No. 1, 1-22 (1997).

[18] L. Shampine, I. Gladwell and S. Thompson, *Solving ODEs with MATLAB* (First Edition, Cambridge University Press, New York, 2003).

[19] <http://www.pdecomp.net/TheCompendium/downloads.php> under the compendium SRC of Chapter 6 directory. (accessed May 27, 2013)

[20] P. Wesseling, *Principles of Computational Fluid Dynamics* (Springer, Berlin, 2001).

[21] C-W. Shu, "Essentially Non-oscillatory and Weighted Essential Non-oscillatory Schemes for Hyperbolic Conservation Laws," in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C-W. Shu, E. Tadmor, Lecture Notes in Mathematics **1697**, 325-432 (Springer, Berlin, 1998).

[22] G. Griffiths, and W. Schiesser, *Traveling Wave Solutions of Partial Differential Equations: Numerical and Analytical Methods with Matlab and Maple* (Academic Press 2011). Available at: <http://www.pdecomp.net/> (accessed November 27, 2012)

[23] <http://www.pdecomp.net/TravelingWaves/downloadsTW.php>

[24] F. Ravet, X. Bao, Y. Li, A. Yale, V. Kolosha, and L. Chen, "Signal processing technique for distributed Brillouin sensing at centimeter spatial resolution," *IEEE Sens. J.* **6**, 3610-3618 (2007).

A APPENDIX

```
function ut=pde(t,u)
%
% Problem parameters

global xl xu n ncall gamma_k
global coeff1 coeff2 coeff3 coeff4 TT1 TT2 omega1 omega2
global dx1 dx2 g1 g2

i = 1:n; % spatial index
i1 = i(dx1<=i & i<=dx2); % 'inner' region
i2 = i(i<dx1 | dx2<i); % 'outer' region
coeff = zeros(n,1);
coeff(i1) = coeff2;
coeff(i2) = coeff1;

coeffsin = zeros(n,1);
coeffsin(i1) = coeff4;
coeffsin(i2) = coeff3;
omega = zeros(n,1);
omega(i1) = omega2;
omega(i2) = omega1;

TT = zeros(n,1);
TT(i1) = TT2;
TT(i2) = TT1;

g = zeros(n,1);
g(i1) = g2;
g(i2) = g1;

% u to y, columnn vectors

u = reshape(u,n,6);
y1 = u(:,1); %As
y2 = u(:,2); %Ap
y3 = u(:,3); %Aa
y4 = u(:,4); %dAa/dt
y5 = u(:,5); %Phi_a
y6 = u(:,6); %dPhi_a/dt

% BCs
y1(1)=pulse(t);
y2(n)=5e-3;
```

```

y1x=-super(xl,xu,n,y1,1)';
y2x=super(xl,xu,n,y2,-1)';

% Derivatives in t
% d(y1)/dt
y1t = y1x - y2.*y3.*coeff;
y1t(1) = 0; % boundary condition

% d(y2)/dt
y2t = y2x + y1.*y3.*coeff;
y2t(n) = 0; % boundary condition

y3t(i)=y4; % y3t= dAa/dt
y5t(i)=y6; % y5t= dPhi_a/dt y6t= d^2Phi_a/dt^2

y4t= (-g*y1*y2*coeffsin-(2*gamma_k*y4+2*y3*omega*y6)+...
      y3*y6*y6 -TT*y3 );

y6t= (-g*y1*y2*coeffsin+2*omega*( y4+ gamma_k*y3)-...
      2*y6*(y4+y3*gamma_k))/y3 ;

% yt to ut, column vector
ut = [y1t;y2t;y3t;y4t;y5t;y6t];

% Increment calls to pde'1
ncall=ncall+1;

```