The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0332-1649.htm

Novel guess functions for efficient analysis of Raman fiber amplifiers

Fikri Serdar Gokhan
Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Gazikent University, Gaziantep, Turkey, and

Gunes Yilmaz
Faculty of Engineering and Architecture, Department of Electronics Engineering, Uludag University, Gorukle, Turkey

Abstract

Purpose – The aim of the paper is to demonstrate a fast numerical solution for Raman fiber amplifier equations using proposed guess functions and MATLAB intrinsic properties. MATLAB BVP solvers are addressed for the solution.

Design/methodology/approach – The guess functions proposed for the solution of RFA equations using MATLAB BVP solvers are derived from Taylor expansion of pump and signal wave near the boundary to specifically obtain convergence for the initial mesh point. The guess functions increase simulation speed significantly. In order to improve the simulation speed further, vectorization and analytical Jacobians are introduced. Comparisons among `bvp4c` and `bvp5c` have been made with respect to total pump power, number of signals, vectorization with/without analytical Jacobians, fiber length, relative tolerance and continuation method. The simulations are performed to determine the effect of the run time on the choice of the number of equally spaced mesh points (N) in the initial guess, and thus optimal N values are found.

Findings – MATLAB BVP solvers have been proven to be effective for the numerical solution of RFAs with the proposed guess functions. In particular, with vectorizing, run time reduction is between 2.1 and 5.4 times for `bvp4c` and between 1.6 and 2.1 times for `bvp5c` and in addition to vectorizing, with the introduction of the analytical Jacobians, the reduction is between 2.4 and 6.2 times for `bvp4c` and 1.7 and 2.2 times for `bvp5c`, respectively, depending on the total pump power between 1,000 mW and 2,000 mW and the number of signals. Also, simulation results show that the efficiency of the solution with proposed guess functions is improved more than six times compared with those of previously reported continuation methods. Results show that the proposed guess functions with the vectorization and analytical Jacobians can be used for the performance evaluation of RFAs for the high power systems/long gain fiber span.

Practical implications – The robust improvement of the solution proposed in this paper lies in the fact that the derived guess functions for the RFAs are highly effective in the sense that they assist the solver to converge to the solution for any total pump power value in a wide range from 1 to 3,000 mW and for any fiber lengths ranging 1 to 200 km which are used in practical applications. Hence, it is practicable for the performance evaluation of the existing RFA networks.

Originality/value – The novelty of this method is that, starting with the co-propagating single pump and signal RFA schema, the authors derived the guess function specifically for the initial mesh points rather than using its analytical approximations. Moreover, the solution is generalized for co-/counter propagating pumps/signals with the curve fitted coefficient(s).

Keywords MATLAB BVP solvers, Guess functions, Raman fiber amplifiers (RFAs), Boundary value problems, Numerical analysis

Paper type Research paper

COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering
Vol. 31 No. 2, 2012
pp. 330-345

0332-1649
DOI 10.1108/03321641211199782

1. Introduction

RFAs, which are widely commercialized nonlinear optical devices in telecommunications, are being used in almost every new long-haul and ultralong-haul fiber-optic transmission system. RFAs have become very attractive for broad bandwidth applications since they improve the noise figure and reduce the nonlinear penalty of fiber systems. Moreover, they allow for longer amplifier spans, higher bit rates, closer channel spacing and operation near the zero-dispersion wavelength (Islam, 2002). The practical demands for and increased research interest in RFAs have implied reasonable and efficient simulation methods for the performance evaluation of RFAs before the real amplifiers are manufactured.

The problem of RFA modeling has been widely discussed for quite some time and numerous models have been presented for RFA simulation (Kidorf *et al.*, 1999; Mandelbaum and Bolshtyanshy, 2003). The mathematical model describing the interactions among the lightwaves in RFA is a nonlinear two-point boundary value problem (BVP). The available numerical methods to solve this RFA propagation equation usually suffer from exhaustive computing run times, especially when the bandwidth of the RFA is wide and the number of the signal channels is large.

The common techniques to solve BVPs can be classified as either shooting methods or finite difference methods. A shooting method starts from an initial guess and treats the problem as an initial value problem (IVP). Then an iteration method is used to correct the initial guess until the boundary conditions are satisfied. Finite difference methods use information from the previous mesh points. Since multistep finite difference methods improve the accuracy of the solution at each step, they can effectively decrease the computing time with the same accuracy (Liu and Zhang, 2004).

Generally IVPs have a unique solution. However, BVP is different from IVP in the sense that it may have no solution or a single solution, or multiple solutions. In order to direct the solver for the solution of interest, it is necessary to assist the solver by informing it with a guess. Computation of the solution of interest and whether any solution is achieved or not depends strongly on the initial guess. Therefore, when solving BVPs the user must provide a guess for the solution, not only to identify the solution of interest but also to assist the solver in computing the desired solution (Kierzenka and Shampine, 2001).

The most difficult part for the solution of BVPs is to provide an initial estimation to the solution. MATLAB BVP solvers call for users to provide guesses for the mesh and solution that will lead to convergence. Although MATLAB BVP solvers take an unusual approach to the control of error in case of having poor guesses for the mesh and solution, especially for the nonlinear BVP, a good guess is necessary to obtain convergence (Shampine *et al.*, 2003). Recently, a continuation method with the simple guess values which are equal to the boundary values has been proposed and demonstrated using MATLAB BVP solvers (Gokhan and Yilmaz, 2009). In this method, once the convergence length is computed, computation for the remaining length is calculated by the continuation process. However, at each step of augmenting the length/power, the computation process is performed once again which increases the computation time. It is natural that if the convergence length is augmented for the whole interval, the computation is performed only once and the simulation time is decreased substantially. Extending the convergence length for whole length of the fiber distance requires efficient guess values/functions. Therefore, for the RFA equations, supplying an efficient guess not only improves the speed of the solution but it is also necessary to achieve the solution of interest.

In order to significantly improve the simulation speed of RFA equations compared with the continuation method, MATLAB BVP solvers with highly efficient guess functions are introduced and numerically demonstrated in this paper. The guess functions are deliberately derived for the initial mesh points, because even when a guess function is supplied that is a good approximation for the solution, the solver uses its values only on the initial mesh (Shampine *et al.*, 2003). By using the co-propagating one pump and one signal Raman amplifier scheme, the guess functions are derived from Taylor expansion of the pump and signal wave near the end points. With the help of the MATLAB Symbolic Toolbox, Taylor series expansions with four terms of pump and signal evolutions near $z = 0$ are obtained. Thereafter, the derived guess functions are curve fitted to the real solution and fitting coefficient is obtained with the help of the MATLAB Curve Fitting Toolbox. The results of this research have demonstrated that for the backward propagation, where the pump is inserted at $z = L$, the same guess functions with the negative sign and different fitting coefficient can be used. The novelty of this method is that, without any necessity for the continuation method, the derivation of the approximate solution for the initial mesh is enough to yield convergence for the whole interval. Numerical results show that compared with the continuation method, by using the proposed guess functions the efficiency is improved between 5.6 and 10 times with `bvp4c` and between 11.2 and 13 times with `bvp5c` depending on the total pump power between 1,000 and 2,000 mW and number of signals. In order to improve the solvers' efficiency and reduce the run time further, we have used MATLAB intrinsic speeding-up properties such as vectorization and supplying analytical Jacobians for evalution of the differential equations. Simulation results show that, with vectorizing, this reduction is between 2.1 and 5.4 times for `bvp4c` and between 1.6 and 2.1 times for `bvp5c` and in addition to vectorizing if the analytical Jacobians are introduced this reduction is between 2.4 and 6.2 times for `bvp4c` and 1.7 and 2.2 times for `bvp5c`, respectively, depending on the total pump power between 1,000 and 2,000 mW and number of signals.

2. Theoretical model

The mathematical model of a multi-pumped RFA includes a large number of effects, the most important of which, for the purposes of the present consideration, pump-pump, signal-pump, signal-signal interactions and fiber attenuation experienced by both pump and signal waves. In the steady state, propagation equations governing the power evolutions of pump and signals in RFA can be expressed as the following systems of nonlinearly coupled differential equations (Perlin and Winful, 2002):

$$\pm\frac{dP_k}{dz} = -\alpha_k P_k + \sum_{j=1}^{k-1}\frac{g(v_{j,k})}{\Gamma A_{eff}}P_jP_k - \sum_{j=k+1}^{m+n}\frac{v_k}{v_j}\frac{g(v_{k,j})}{\Gamma A_{eff}}P_jP_k \quad k = 1,2,3,\ldots,n+m \quad (2.1)$$

The minus and plus symbols denote the backward-propagating pump waves and forward-propagating signal waves, respectively. The frequencies are numerated in decreasing order ($v_k > v_j$ for $i < j$), indexes $k = 1,\ldots,n$ correspond to the backward-propagating pump waves and indexes $k = n + 1,\ldots, n + m$ correspond to the forward-propagating signal waves. Here, values P_k, v_k and α_k describe, respectively, the power, frequency and attenuation coefficient for the kth wave, $k = 1, 2,\ldots, n + m$. $g(v_{j,k})$ is the Raman gain coefficient from wave j to k measured with a pump at frequency v_{ref}, that is, $g(v_{j,\,k}) = v_j/v_{ref} \times g_R\,(v_j - v_k)$ and $g(v_{k,\,j})$ is the Raman gain coefficient from wave k to j measured with a pump at frequency v_{ref}, that is, $g(v_{k,j}) = v_k/v_{ref} \times g_R\,(v_k - v_j)$. The frequency ratio v_k/v_j denotes vibrational losses. A_{eff} is the effective area of optical fiber and Γ is the polarization factor, the value which lies between 1 and 2. For a backward-pumped amplifier with length L, signal beams are inserted at $z = 0$ and pump beams inserted at $z = L$. The boundary conditions for this system are:

$$P_P(z = L) = P_L \tag{2.2}$$

$$P_S(z = 0) = P_0 \tag{2.3}$$

where P_P and P_S show the power of the pumps and signals, respectively. In this system we have the power of signals at $z = 0(P_0)$ and the power of pumps at $z = L(P_L)$.

3. The proposed guess functions and implementation

Estimates are provided for both the solution and the mesh. For the solution, the guess may be the value or the function. It must be emphasized that the solver uses its values only on the initial mesh even though it may have used a guess function. Therefore, if the estimation for the initial mesh is calculated, the solver is able to compute the solution for the subsequent mesh points.

3.1 Derivation of guess functions

In order to estimate the guess function for the distributed multi-pumped (RFA) equations, forward-pumping RFA configuration with single pump and signal can be exploited. In this configuration, the differential equation system describing the intensities of signal and pump waves under the influence of Raman scattering during their propagation through the medium are represented by:

$$\frac{d}{dz} I_S = \frac{g_R}{\Gamma} I_P I_S - \alpha_S I_S \tag{3.1a}$$

$$\frac{d}{dz} I_P = -\frac{\omega_P}{\omega_S} \frac{g_R}{\Gamma} I_P I_S - \alpha_P I_P \tag{3.1b}$$

where I_p and I_s describe intensity of pump and signal, respectively; g_R is Raman gain; ω_P and ω_S are pump and signal wavelengths, respectively; α_P and α_S are attenuation constants for pump and signal waves, respectively. Γ is a factor that includes the relative polarization between the pump and stokes waves.

The boundary conditions for this system are:

$$P_P(z = 0) = P_p(0) \tag{3.2a}$$

$$P_S(z = 0) = P_S(0) \tag{3.2b}$$

where $P_P(0)$ and $P_S(0)$ show the initial power of the pumps and signals, respectively.

In order to find the estimation for the initial mesh we may look for the analytical approximations in the form of Taylor series near the left boundary which is $z = 0$ and then generalize this guess for $z = $ L. Exploiting the MATLAB Symbolic Toolbox functionality, we use the following script to substitute four terms of Taylor series expansions into equations (3.1).

```
%= = = = = = = = = = = = = = = = = = = = = = = = = = = = =
syms z A B C D E F Is Ip gr gamma wp ws alpha_s alpha_p Is0 Ip0
Eqn1 Eqn2 Eqn3
Is = Is0 + A*z + B*z^2 + C*z^3;
Ip = Ip0 + D*z + E*z^2 + F*z^3;
Eqn1 = collect (diff(Is,'z') - (gr/gamma)*Ip*
Is + alpha_s*Is)
Eqn2 = collect (collect (diff (Ip,'z') + (wp/ws)*(gr/gamma)*
Ip*Is + alpha_p*Ip), z)
% gamma = Γ, alpha_s = α_S, alpha_p = α_p, Is0 = Is(0),
Ip0 = Ip(0)
%= = = = = = =  = = = = = =  = = = = = =  = = = = = =  = =
```

It must be emphasized that we have taken into account the initial/boundary conditions (3.2). This script produces the output:

Eqn1 =

$$
\begin{aligned}
&-g_R/\Gamma*F*C*z\hat{}6+(-g_R/\Gamma*E*C-g_R/\Gamma*F*B)*z^5+\cdots\\
&(-g_R/\Gamma*D*C-g_R/\Gamma*E*B-g_R/\Gamma*F*A)*z^4+\cdots\\
&(\alpha_S*C-g_R/\Gamma*I_p(0)*C-g_R/\Gamma*D*B-g_R/\Gamma*E*A\\
&-g_R/\Gamma*F*Is(0))*z^3+\cdots\\
&(-g_R/\Gamma*Ip(0)*B-g_R/\Gamma*D*A-gr/\Gamma*E*Is(0)+3*C+\alpha_S*B)*z^2+\cdots\\
&(2*B-g_R/\Gamma*Ip(0)*A-g_R/\Gamma*D*IS(0)+\alpha_S*A)*z\\
&+A+\alpha_S*IS(0)-g_R/\Gamma*Ip(0)*Is(0)
\end{aligned}
\tag{3.3a}
$$

Eqn2 =

$$
\begin{aligned}
&w_p/w_s*g_R/\Gamma*F*C*z\hat{}6+(w_p/w_s*g_R/\Gamma^*E^*C+w_p/w_s*gr/\Gamma*F*B)*z^5+\\
&w_p/w_s*g_R/\Gamma*D*C+w_p/w_s*g_R/\Gamma*E*B+w_p/w_s*g_R/\Gamma*F*A)*z^4+\cdots\\
&(\alpha_P*F+w_p/w_s*g_R/\Gamma*IP(0)*C+w_p/w_s*g_R/\Gamma*D*B\\
&+w_p/w_s*g_R/\Gamma*E*A+w_p/w_s*g_R/\Gamma*F*IS(0))*z^3+\cdots\\
&(w_p/w_s*g_R/\Gamma*Ip(0)*B+w_p/w_s*g_R/\Gamma*D*A\\
&+w_p/w_s*g_R/\Gamma*E*IS(0)+3*F+\alpha_P*E)*z^2+\cdots\\
&(2*E+w_p/w_s*g_R/\Gamma*IP(0)*A+w_p/w_s*g_R/\Gamma*D*IS(0)+\alpha P*D)*z+\cdots\\
&D+\alpha_P*IP(0)+w_p/w_s*g_R/\Gamma*Ip(0)*Is(0)
\end{aligned}
\tag{3.3b}
$$

We are interested in the behavior as $z \to 0$ and so, the higher the power of z, the less effect it has in these expansions. Our goal is to satisfy the equations as well as possible, so we want to choose coefficients that make as many successive terms zero as possible, starting with the lowest power. To eliminate the constant terms, we see from the expansions that we must take:

$$\mathbf{A} = -IS(0)*(\alpha_s*\Gamma - g_R*I_P(0))/\Gamma \tag{3.4a}$$

$$\mathtt{D} = -\mathtt{IP}(0) * (\alpha_p * w_s * \Gamma + w_p * g_R * \mathtt{I_S}(0))/w_s/\Gamma \tag{3.4b}$$

Next, to eliminate in terms in z we must take:

$$\mathtt{B} = -1/2 * (-g_R * \mathtt{IP}(0) * \mathbf{A} - g_R * \mathbf{D} * \mathtt{IS}(0) + \alpha_s * \mathbf{A} * \Gamma)/\Gamma \tag{3.5a}$$

$$\mathbf{E} = -1/2 * (w_p * g_R * \mathtt{IP}(0) * \mathbf{A} + w_p * g_R * \mathbf{D} * \mathtt{IS}(0) * \alpha_p * \mathbf{D} * w_s * \Gamma)/w_s/\Gamma \tag{3.5b}$$

Next, to eliminate in terms in z^2 we must take:

$$\mathbf{C} = 1/3 * (g_R * \mathtt{Ip}(0) * \mathbf{B} + g_R * \mathbf{D} * \mathbf{A} + g_R * \mathbf{E} * \mathtt{IS}(0) - \alpha_s * \mathbf{B} * \Gamma)/\Gamma \tag{3.6a}$$

$$\begin{aligned}\mathbf{F} = -1/3 * (&w_p * g_R * \mathrm{Ip}(0) * \mathbf{B} + w_p * g_R * \mathbf{D} * \mathbf{A} \\ &+ w_p * g_R * \mathbf{E} * \mathtt{IS}(0) + \alpha_p * \mathbf{E} * w_s * \Gamma)/w_s/\Gamma\end{aligned} \tag{3.6b}$$

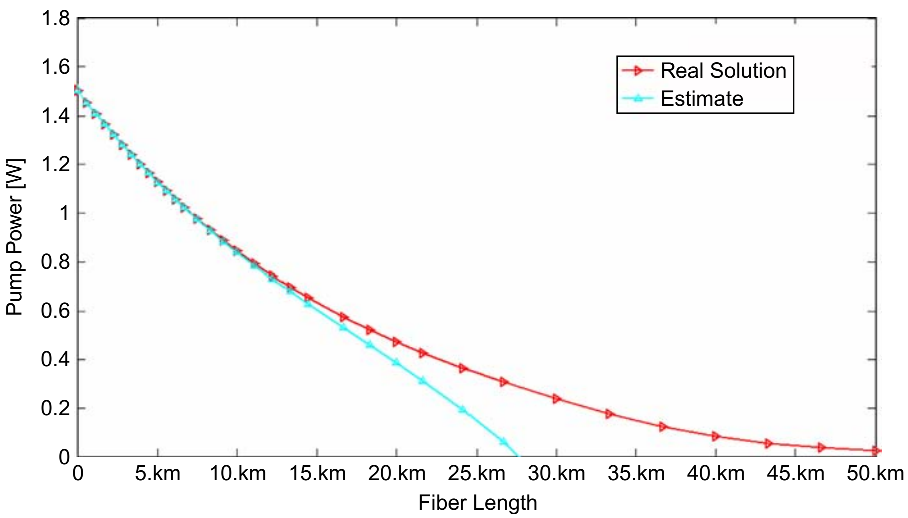
We thus conclude that, for small values of z, we have:

$$\mathbf{IS}(z) \approx \mathtt{IS0} + \mathbf{A} * z + \mathbf{B} * z^2 + \mathbf{C} * z^3 \tag{3.7a}$$

$$\mathbf{IP}(z) \approx \mathtt{IP0} + \mathbf{D} * z + \mathbf{E} * z^2 + \mathbf{F} * z^3 \tag{3.7b}$$

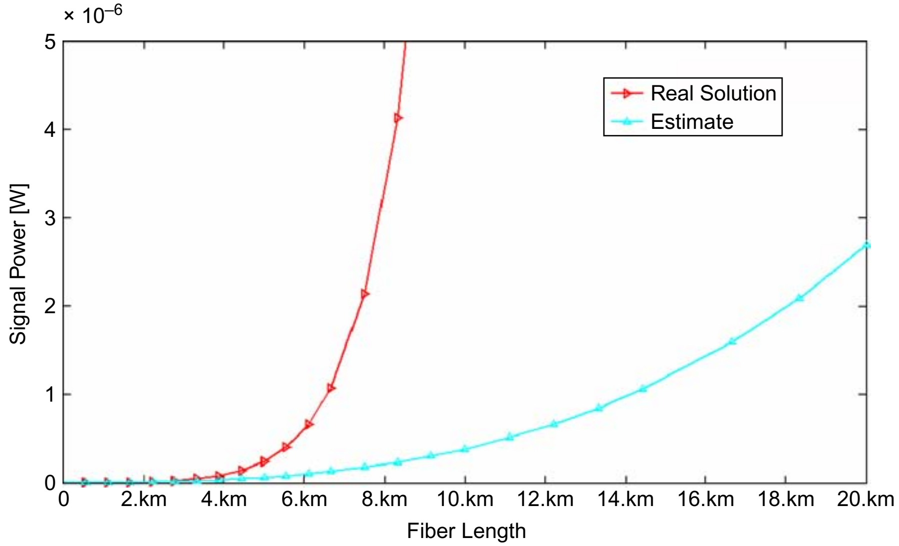
Figures 1 and 2 show the comparison of real and estimated solutions for pump and signal power evolution, respectively.

In Figure 1 the first 15 km and in Figure 2 the first 4 km, the real solution exactly matches with the Taylor series expansion. Therefore, if the first four terms of a Taylor expansion is used as guess functions for the FRA equations, they will be adequate for the computation of not only the initial mesh points but also the following ones.



Notes: P_{s0} = 1 nW, α_p = 0.25 dB/km, α_s = 0.2 dB/km, $g_R = 7 \times 10^{-14}$ m/W, A_{eff} = 80 μm^2, ω_p = 1.455 μm, ω_s = 1.555 μm, L = 50 km, Γ = 1

Figure 1.
Comparison of real solution and estimated solution of pump power for the forward-pumping RFA configuration with single pump and signal



Notes: P_{s0} = 1 nW, α_p = 0.25 dB/km, α_s = 0.2 dB/km, $g_R = 7 \times 10^{-14}$ m/W, A_{eff} = 80 μm^2, ω_p = 1.455 μm, ω_s = 1.555 μm, L = 50 km, Γ = 1

Figure 2.
Comparison of real solution and estimated solution of signal power for the forward-pumping RFA configuration with single pump and signal

However, in order to determine the guess functions for multi-pumped RFAs, real solution of the pump and signal curves are fitted to the Taylor series expansion curves of single pumped RFA, and fitting coefficient is determined. By the help of the MATLAB Curve Fitting Toolbox, the fitting coefficient is estimated as $K_1 \approx 10^{-2}$. Therefore, the guess functions for RFA will be:

$$P_S(z) \approx K_1 \cdot I_S(z) \cdot A_{eff} \approx K_1 \cdot (I_S(0) + A \cdot z + B \cdot z^2 + C \cdot z^3) \cdot A_{eff} \tag{3.8a}$$

$$P_P(z) \approx K_1 \cdot I_p(z) \cdot A_{eff} \approx K_1 \cdot (I_p(0) + D \cdot z + E \cdot z^2 + F \cdot z^3) \cdot A_{eff} \tag{3.8b}$$

Equations (3.8a) and (3.8b) are used for the forward-pumping RFA configuration. For the backward-pumping RFA configuration when the pump wave is inserted from $z = L$, the sign of the equation (3.8a) remains same but the sign of the equation (3.8b) will be the opposite and the fitting coefficient for backward wave (here pump) is estimated as $K_2 \approx 10^{-1}$:

$$P_S(z) \approx K_1 \cdot I_S(z) \cdot A_{eff} \approx K_1 \cdot (I_S(0) + A \cdot z + B \cdot z^2 + C \cdot z^3) \cdot A_{eff} \tag{3.9a}$$

$$P_P(z) \approx -K_2 \cdot I_p(z) \cdot A_{eff} \approx -K_2 \cdot (I_p(0) + D \cdot z + E \cdot z^2 + F \cdot z^3) \cdot A_{eff} \tag{3.9b}$$

Therefore, $K_1 \approx 10^{-2}$, for waves which are propagating $z = 0$ to z = L and $K_2 \approx 10^{-1}$, for waves which are propagating $z = L$ to $z = 0$.

3.2 Coding guess functions

The guess is supplied to MATLAB BVP solvers using the auxiliary function `bvpinit`. This function accepts the guess structure using two arguments. The first argument of the guess deals with supplying a mesh structure that reveals the behavior of the solution. The second argument of the structure deals with supplying the guessed values of the solution or the function for computing the guessed values of the solution on the mesh that is specified with the first argument of the guess structure. For example:

$$\texttt{solinit} = \texttt{bvpinit(linspace(0,L,}\mathbf{N}\texttt{),@guess);} \quad (3.10)$$

Here N equally spaced points in [0, L] (L is the fiber length) are tried and the guess is provided by means of a function guess. In Section 3.3, the guess function with a script for the backward-pumping RFA configuration including ten pumps and 80 signals is illustrated.

3.3 Script for the guess functions

```
%= = = = = = =  = = = = = = =  = = = = = = =  = = = = = = = =
  function v = guess(x)
  global Ip0 Is0 Aeff gamma wp ws alpha_s alpha_p
  % For Backward pumping RFA configuration (counter propagating
  pump and signals)
  % gamma = Γ, alpha_s = αS, alpha_p = αp, Is0 = Is(0), Ip0 =
  Ip(0)
  K1 = 1e-2;
  K2 = 1e-1;
  A = -Is0*(alpha_s*gamma-gr*Ip0)/gamma
  D = -Ip0*(alpha_p*ws*gamma + wp*gr*Is0)/ws/gamma
  E = -1/2*(wp*gr*Ip0*A + wp*gr*D*Is0 + alpha_p*D*ws
  *gamma)/ws/gamma
  B = -1/2*(-gr*Ip0*A-gr*D*Is0 + alpha_s*A*gamma)/gamma
  C = 1/3*(gr*Ip0*B + gr*D*A + gr*E*Is0-alpha_s*B*gamma)/
  gamma
  F = -1/3*(wp*gr* Ip0*B + wp*gr* D*A + wp*gr* E*Is0 + alpha_
  p*E*ws*gamma)/ws/gamma
  Guess_for_signals = K1*(Is0 + A*x + B*
  x^2 + C*x^3)*Aeff;
  Guess_for_pumps = -K2*(Ip0 + D*x + E*x^2 + F*x^3)*Aeff;
  % For Forward-pumping RFA configuration (co-propagating pump
  and signals)
  % Guess_for_signals = K1*(Is0 + A*x + B*x^2 + C*x^3)*Aeff;
  % Guess_for_pumps = K1*(Ip0 + D*x + E*x^2 + F*x^3)*Aeff;
  v = zeros(90,1);
  v(1:10) = Guess_for_pumps;
  v(11:90) = Guess_for_signals;
  %= = = = = = =  = = = = =  = = = = = =  = = = = = =  = = = = =
```

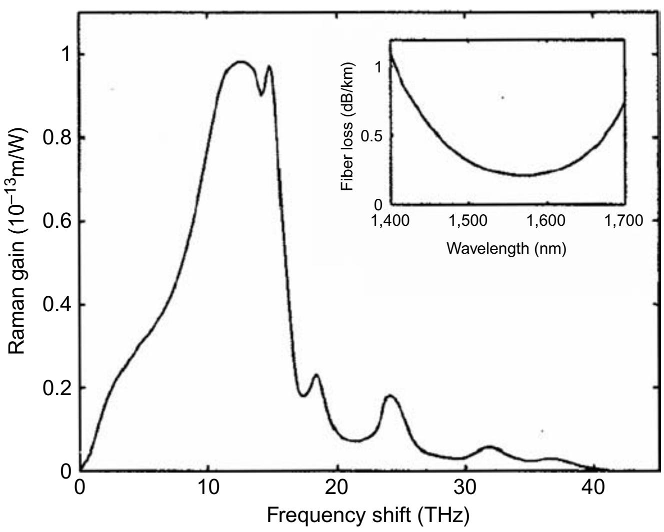
4. Effectiveness and convergence

In order to verify the convergence of the proposed guess functions, simulations are performed with ten pumps and 80 signals. In the simulations, it is assumed that the signals range from 189.4 to 197.4 THz (1,519-1,583 nm) with a power of 0.5 mW/channel: $A_{eff} = 80\,\mu m^2$, $\Gamma = 2$, $L = 60$ km and $v_{ref} = 196.4$ THz (1,511 nm).

Depending on the wavelength, fiber losses vary between $\alpha = 0.20\text{-}0.23$ dB/km for signals and $\alpha = 0.26\text{-}0.23$ dB/km for pumps, respectively, (Figure 3). The wavelengths of the pumps are 1,415, 1,421, 1,430, 1,435, 1,442, 1,450, 1,463, 1,470, 1,475 and 1,499 nm. The proposed guess functions are used for two pumping configurations, backward and bi-directional. For the simulation of backward pumping configuration in Figure 4, all the pumps are backward propagating and their powers are spaced between 70 and 320 (total power is 2,000 mW). For the simulation of bi-directional pumping configuration, in Figure 5 seven forward-propagating and three backward-propagating pumps are used; their powers are spaced between 70 and 320 mW (total power is 2,000 mW). The wavelengths of the forward-propagating pumps are 1,415, 1,421, 1,450, 1,463, 1,470, 1,475 and 1,499 nm and backward-propagating pumps are 1,430, 1,435 and 1,442 nm. Table I shows the pump powers at the beginning and end of the fiber for backward and bi-directional pumping, respectively.

In order to verify the proposed guess method and to demonstrate its performance improvement with respect to total pump powers and fiber length, a series of RFAs with different total pump powers and fiber lengths are simulated. Numerical simulations are performed to prove that the guess functions which are derived from the Taylor series expansions are accurate enough to ensure the convergence for any total pump power value in a wide range from 1 to 3,000 mW and for any fiber length ranging 1-200 km. Consequently, the proposed guess functions can be used for the performance evaluation of RFAs for the high power systems/long gain fiber span with forward, backward, or bi-directional pumping configurations. The calculations are performed in MATLAB 7.5 on a personal laptop computer with an Intel Centrino Duo 1.83 GHz processor.

The simulations are performed for the effect of the run time on the choice of the number of equally spaced mesh points (N) in the initial guess (3.10). In these



Note: Inset shows the linear attenuation spectrum of a silica fiber in the low-loss window 1,400-1,700 nm

Figure 3.
Typical Raman gain spectrum $g_R\ (\Delta v)$ of a silica fiber

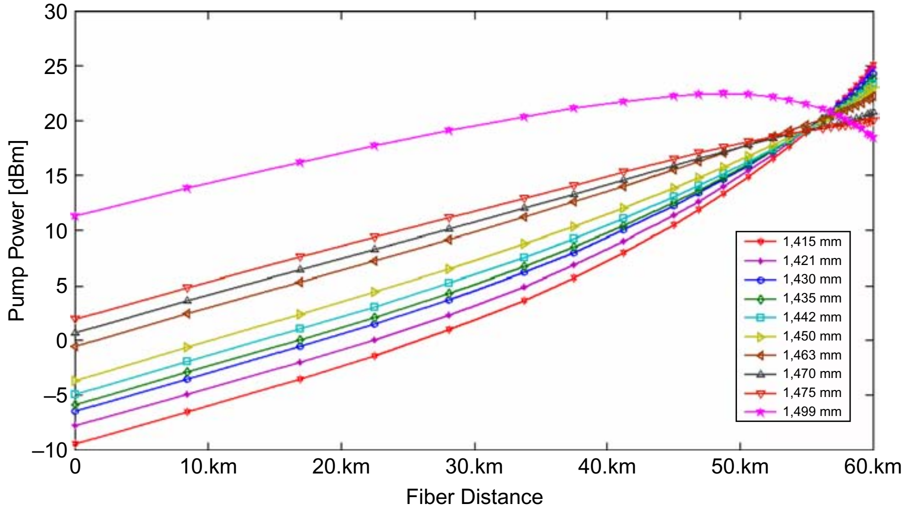


Figure 4.
Power evolution of the backward-propagating pumps along the fiber

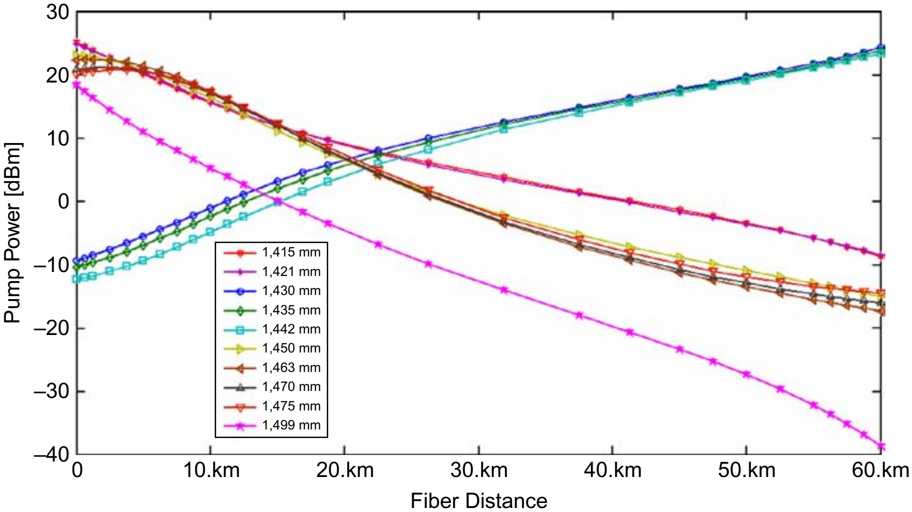


Figure 5.
Power evolution of the bi-directional propagating pumps along the fiber for seven forward and three backward pumps at different wavelength and powers

simulations, four pump powers are considered. All the backward-propagating pumps have an equal power of 100, 130, 170 and 200 mW (total power 1,000, 1,300, 1,700 and 2,000 mW), respectively. For each pump power level, the simulations are performed with 20, 40, 60 and 80 signals.

When the number of initial mesh points less than required is specified, the solvers failed because the behavior of the solution is not revealed on a mesh of so few points. Thereafter, the mesh points are increased step by step until convergence is accomplished. However, it must be emphasized that, in general, increasing the initial mesh points is accompanied by increasing run time. Consequently, when the mesh

Table I.
Pump powers for backward and bi-directional configuration

Set 1 (total input pump power is 2,000 mW)		Set 2 (seven forward and three backward pumps – total input pump power is 2,000 mW)	
P_p ($z = 0$) (mW)	P_S ($z = L$) (mW)	P_p ($z = 0$) (mW)	P_S ($z = L$) (mW)
0.1139	320 (1,415 nm) backward	320 (1,415 nm) forward	0.1329
0.1665	300 (1,421 nm) backward	300 (1,421 nm) forward	0.1393
0.2272	265 (1,430 nm) backward	0.1143	265 (1,430 nm) backward
0.2595	240 (1,435 nm) backward	0.0911	240 (1,435 nm) backward
0.3211	215 (1,442 nm) backward	0.058	215 (1,442 nm) backward
0.4322	200 (1,450 nm) backward	200 (1,450 nm) forward	0.030
0.8886	170 (1,463 nm) backward	170 (1,463 nm) forward	0.018
1.177	120 (1,470 nm) backward	120 (1,470 nm) forward	0.024
1.572	100 (1,475 nm) backward	100 (1,475 nm) forward	0.035
13.38	70 (1,499 nm) backward	70 (1,499 nm) forward	0.000135

points (N) more than the required are specified the run time is increased. Therefore, in order to achieve the shortest run time, the value of N has to be optimized. The optimal number of initial mesh points (N) is where the solver has just converged to the solution robustly or one or two points above. With respect to the simulations for the given parameters, the optimal number of N is found between 4 and 7. Table II shows the optimal N values related with the simulation parameters.

5. Reduction of the run time with MATLAB

The first technique which is used to reduce run time is vectorizing the evaluation of the differential equations. Vectorization is a valuable tool for speeding up `MATLAB` programs and this greatly reduces the run time (Shampine *et al.*, 2003). By vectorization, the function $f(x, y)$ is coded so that when given a vector `x = [x1, x2, ...]` and a corresponding array of column vectors `y = [y1, y2, ...]`, it returns an array of column vectors `[f(x1, y1), f(x2, y2), ...])`. By default, `bvp4c` and `bvp5c` approximate a Jacobian using finite differences. If the evaluation of the Raman propagation equations is vectorized, the computation of the approximate Jacobian is relieved and the run time is often greatly reduced. The evaluation of the RFA equations is vectorized by changing the vectors to arrays and changing the multiplication to an array multiplication. It can be coded by changing scalar quantities like y(1) into arrays like y(1,:) and changing from scalar operations to array operations

Table II.
Optimal N values

	Optimal N values							
	`bvp4c` Number of signals				`bvp5c` Number of signals			
mW	20	40	60	80	20	40	60	80
100	4	5	5	6	4	5	5	6
130	5	5	7	7	5	6	6	7
170	6	6	6	6	6	6	6	6
200	6	6	6	6	6	6	6	6

by replacing * and ^ with .* and .^, respectively. When vectorizing the Raman propagation equations, the solver must be informed about the presence of vectorization by means of the option "Vectorized", "on":

```
options = bvpset('Stats','on','RelTol',1e-3,'Vectorized','on');
```

The vectorization with the piece of code is illustrated in Section 5.1.

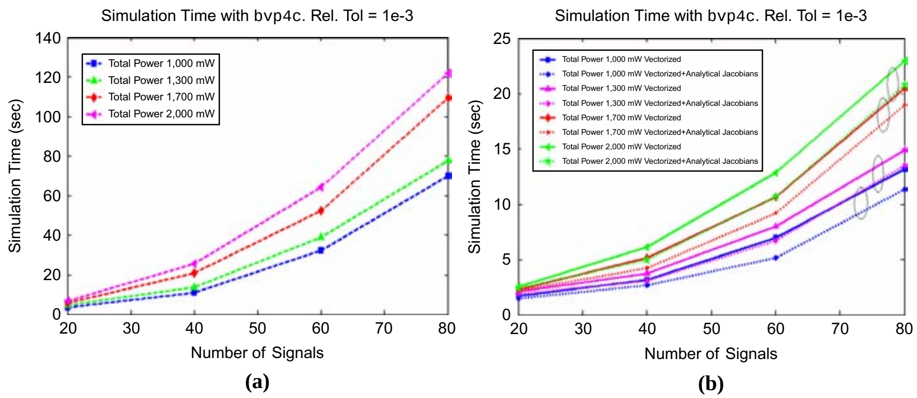
5.1 Illustration of vectorization with the piece of code

```
%= = = = = = =  = = = = = = =  = = = = = =  = = = = = = =
  function dydx = bvp_nonvectorized_ode(x,y)
  global k12(1,4) k12(2,4)
  % Before vectorization
  dydx = [k12(1,4)*y(1)*y(4) + k12(2,4)*y(2)*y(4)]; end
  % After vectorization
  function dydx = bvp_vectorized_ode(x,y)
  global k12(1,4) k12(2,4)
  dydx = [k12(1,4).*y(1,:).*y(4,:) + k12(2,4).*y(2,:).
  *y(4,:)]; end
  %= = = = =  = = = =  = = = =  = = =  = = =  = = = =
```

The second technique is that of supplying analytical partial derivatives or to supply a function for evaluating the Jacobian matrix. This is because, in general, BVPs are solved much faster with analytical partial derivatives. However, this is not an easy task for 80 signal and 10 pumps which require a 90×90 analytical partial derivative matrix. Therefore, for the RFA equations it is less preferable since it is too much trouble and inconvenient, although MATLAB Symbolic Toolbox can be exploited when obtaining analytical Jacobians. The third technique is to supply analytical partial derivatives for the boundary conditions. However, it has less effect on the computation time compared with supplying analytical Jacobians and vectorization. The solver permits the user to supply as much information as possible. It must be emphasized that supplying more information for the solvers results in a shorter computation run time.

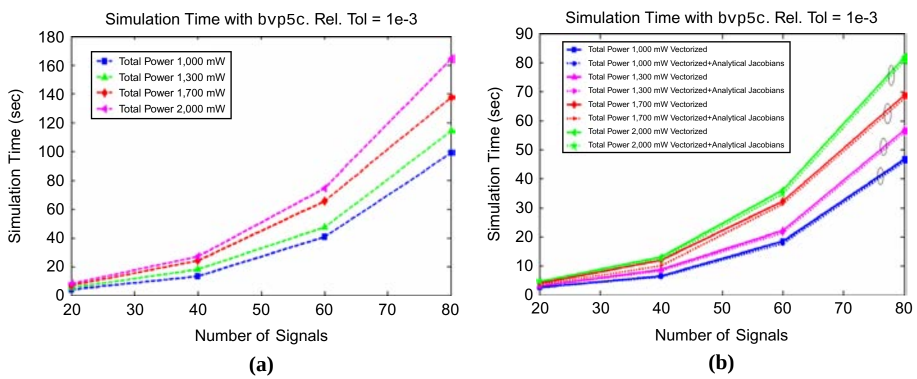
Figures 6 and 7 show the simulation time as a function of the number of signals for `bvp4c` and `bvp5c`, respectively. Figures 6(a) and 7(a) show the simulation time without vectorization and without the introduction of analytical Jacobians. Figures 6(b) and 7(b) show the simulation time when both techniques are used. Figure 8 shows the efficiency of vectorization with/without the introduction of analytical Jacobians for `bvp4c` and `bvp5c`, respectively. From the figures, it can be seen that simulation time is reduced with the introduction of analytical Jacobians and vectorization. Simulation results show that, with vectorizing, this reduction is between 2.1 and 5.4 times for `bvp4c` and between 1.6 and 2.1 times for `bvp5c` and in addition to vectorizing if the analytical Jacobians are introduced this reduction is between 2.4 and 6.2 times for `bvp4c` and between 1.7 and 2.2 times for `bvp5c`, respectively, depending on the total pump power between 1,000 and 2,000 mW and number of signals.

Figure 9 shows the comparison of the simulation times with proposed guess functions and the continuation method proposed by Gokhan and Yilmaz (2009). Apparently, by using the proposed guess functions the efficiency is improved between 5.6 and 10 times with `bvp4c` and between 11.2 and 13 times with `bvp5c` depending



Notes: (a) Without; (b) with vectorization and introduction of analytical Jacobians

Figure 6.
Simulation time as a function of the signal counts with the MATLAB bvp4c solver



Notes: (a) Without; (b) with vectorization and introduction of analytical Jacobians

Figure 7.
Simulation time as a function of the signal counts with the MATLAB bvp5c solver

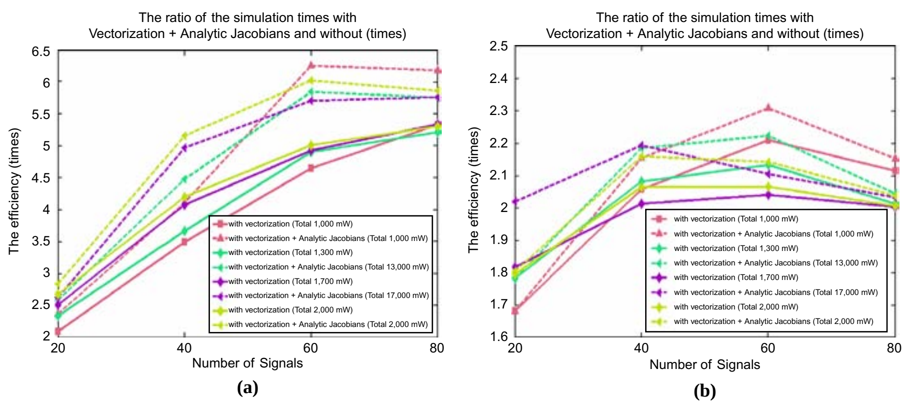


Figure 8.
Efficiency of run time with vectorization and/or introduction of analytical Jacobians with the
(a) MATLAB bvp4c and
(b) MATLAB bvp5c solver

on the total pump power between 1,000 and 2,000 mW and number of signals. With the proposed guess functions the convergence length proposed by Gokhan and Yilmaz (2009) is augmented for the whole interval $[0, L]$ and this makes the continuation method unnecessary.

In order to analyze the effect of the fiber length and relative tolerance (Rel. Tol.) on the run time, simulations are performed using the `bvp4c` solver. In the simulations, the RFA equations are vectorized and 60 signals with 2,000 mW total power are used. Figure 10(a) shows the simulation time as a function of the fiber length. Figure 10(b) shows the simulation time as a function of the relative tolerance.

Figure 10(a) shows that the simulation time grows linearly with the fiber length and in Figure 10(b) it can be seen that the simulation time does not change with relative tolerance value. We believe that there are two main reasons for this. The first reason is that the solver is able to compute the solution with the same number of mesh points

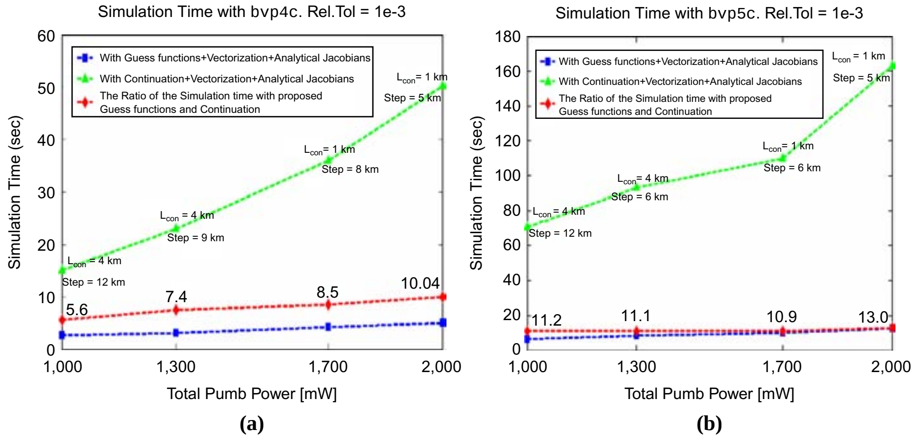
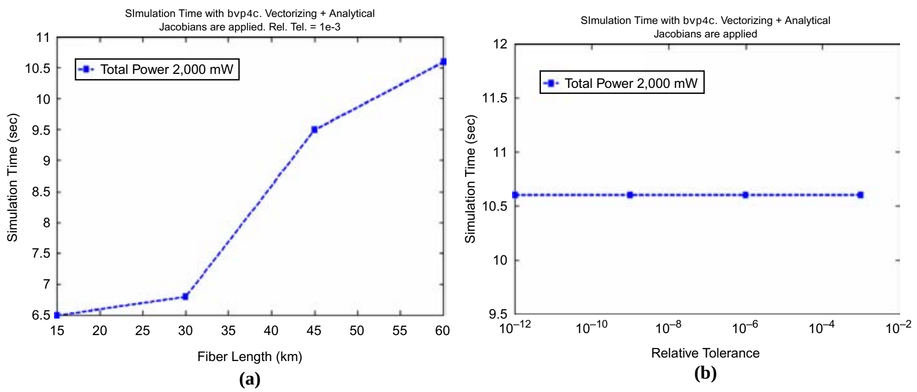


Figure 9. Comparison of the solutions with proposed guess functions and continuation method proposed by Gokhan and Yilmaz (2009) in terms of simulation time with the (a) MATLAB `bvp4c` and (b) MATLAB `bvp5c` solver



Notes: For both figures, ten pump powers with 200mW each, and 60 signals are used; the simulations are performed using `bvp4c` and RFA equations are vectorized and analytical Jacobians are introduced

Figure 10. (a) Simulation time as a function of fiber length and (b) simulation time as a function of relative tolerance

with the relative tolerance between 10^{-12} and 10^{-3}. The second reason is that the proposed guess functions are good enough to assist the solver to compute the solution using the same number of mesh points with stringent relative tolerance values.

6. Conclusion

This paper has proposed and demonstrated the use of efficient guess functions for MATLAB BVP solvers in order to effect a significant improvement in the simulation speed of RFA equations. Exploiting the fact that the MATLAB BVP solvers use the guess values only on the initial mesh, guess functions are derived from the Taylor expansion of the pump and signal wave near the boundary. The efficiency of the solution with guess functions is mainly improved by the use of vectorization and the introduction of analytical Jacobians. Thus, the most time-consuming calculation of the Jacobian matrix has been dramatically relieved. In particular, with vectorizing, run time reduction is between 2.1 and 5.4 times for `bvp4c` and between 1.6 and 2.1 times for `bvp5c` and in addition to vectorizing, with the introduction of the analytical Jacobians the reduction is between 2.4 and 6.2 times for `bvp4c` and 1.7 and 2.2 times for `bvp5c`, respectively, depending on the total pump power between 1,000 and 2,000 mW and the number of signals. Also, simulation results show that the efficiency of the solution with proposed guess functions is improved more than six times compared with those of previously reported methods. These guess functions using vectorization and analytical Jacobians can be used for the performance evaluation of distributed multi-pumped RFAs in the design of forward, backward and bi-directional RFAs for high power/long gain fiber spans.

References

Gokhan, F.S. and Yilmaz, G. (2009), "Solution of Raman fiber amplifier equations using MATLAB BVP solvers", *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 19 December.

Islam, M.N. (2002), "Raman amplifiers for telecommunications", *IEEE J. Sel. Top. Quantum Electron.*, Vol. 8 No. 3, pp. 548-59.

Kidorf, H., Rottwitt, K., Nissov, M., Ma, M. and Rabarijaona, E. (1999), "Pump interactions in a 100-nm bandwidth Raman amplifier", *IEEE Photon. Technol. Lett.*, Vol. 11 No. 5, pp. 530-2.

Kierzenka, J. and Shampine, L.F. (2001), "A BVP solver based on residual control and the MATLAB PSE", *ACM TOMS*, Vol. 27 No. 3, pp. 299-316.

Liu, X. and Zhang, M. (2004), "An effective method for two-point boundary value problem in Raman amplifier propagation equations", *Opt. Commun.*, Vol. 235, pp. 75-82.

Mandelbaum, I. and Bolshtyanshy, M. (2003), "Raman amplifier model in single mode optical fiber", *IEEE Photon.Technol. Lett.*, Vol. 15 No. 12, pp. 1704-6.

Perlin, V.E. and Winful, H.G. (2002), "Optimal design of flat-gain wide-band fiber Raman amplifiers", *Journal of Lightwave Technology*, Vol. 20, pp. 250-4.

Shampine, L.F., Gladwell, I. and Thompson, S. (2003), *Solving ODEs with MATLAB*, 1st ed., Cambridge University Press, New York, NY.

Further reading

Dastjerdi, Z.L., Kroushavi, F. and Rahmani, M.H. (2008), *Optics and Lasers Tech.*, Vol. 40 No. 8, pp. 1041-6.

Han, Q., Ning, J., Zhang, H. and Chen, Z. (2006), *Journal of Lightwave Technology*, Vol. 24 No. 4, pp. 1946-52.

Hu, J., Marks, B.S., Zhang, Q. and Menyuk, C.R. (2005), *J. Opt. Soc. Am. B*, Vol. 22 No. 10, pp. 2083-90.

Kierzenka, J. and Shampine, L.F. (2008), *Journal of Numerical Analysis, Industrial and Applied Mathematics* (*JNAIAM*), Vol. 3 Nos 1/2, pp. 27-41, available at: www.jnaiam.org/downloads.php?did=44 (accessed 5 March 2010).

Ning, J., Han, Q., Chen, Z., Li, J. and Li, X. (2004), *Chin. Phys. Lett.*, Vol. 21 No. 11, pp. 2184-7.

Corresponding author
Fikri Serdar Gokhan can be contacted at: fsgokhan@gmail.com