Nonlinear coupled mode equations on a finite interval: a numerical procedure

C. Martijn de Sterke,[†1] Kenneth R. Jackson[‡] and B. D. Robert[†1]

[†] Department of Physics, University of Toronto

Toronto, Ontario M5S 1A7, Canada

[‡] Department of Computer Science, University of Toronto

Toronto, Ontario M5S 1A4, Canada

We present a numerical procedure to solve a set of nonlinear coupled mode equations on a finite interval. These equations originally arose in the study of the dynamics of gap solitons in nonlinear periodic media. Our procedure, which makes use of an implicit 4^{th} order Runge-Kutta method, is easy to implement, versatile, and very well suited for vectorization or parallelization.

Published in the J. Optical Society of America B,

Vol. 8, No. 2, Feb. 1991, pp. 403–412.

keywords: numerical methods, nonlinear optics.

[1] Also with: Ontario Laser and Lightwave Research Centre

1. Introduction

Coupled mode theory is a powerful tool in the analysis of optical systems and devices.[1,2] It has been applied successfully to optical fiber and waveguide geometries to describe phenomena as diverse as directional coupling, Bragg reflection, as well as electro-optic and acousto-optic mode coupling.[1,2] In using this theory one first defines an "ideal" geometry and determines its associated set of modes, which, by definition, do not interact in the ideal geometry. This is no longer true when perturbations are introduced, and coupled mode theory provides the tools to quantify this interaction. In general this leads to an infinite set of coupled partial differential equations. Often, however, phase matching conditions cause just a few modes to interact substantially, reducing the complexity considerably: In describing the effect of a periodic perturbation, for example, only the forward propagating mode and its Bragg reflected counterpart have to be considered, while all other modes can be ignored, thus giving rise to just two coupled equations.

The deviations from the ideal geometries, and thus the nature of the mode interaction, can be very diverse. The deviations can be periodic, giving rise to Bragg reflection as mentioned above, or can be due to the proximity of another guiding structure, as in directional coupling, or can arise from absorption. Since these perturbations are linear in nature, they give rise to (a set of coupled) *linear* equations. When the perturbation depends on the intensity however, such as when part of the optical system exhibits the Kerr effect, or when the absorption is saturable, the coupled mode equations are nonlinear. Such equations arise for example in describing nonlinear directional couplers,[3] nonlinear periodic media,[4,5] transverse instabilities in nonlinear slabs,[6] and polarization coupling in optical fibers.[7,8] Often one is just interested in the stationary states of these nonlinear systems.[3,9] The dynamical properties of some nonlinear systems have been studied as well, for example pulse propagation through optical fibers.[7,8]

In our studies of nonlinear periodic media we have had the need to study the dynamics of geometries of *finite* extent.[4,5] In the present paper we describe a finite difference method suitable for such systems. It makes use of the characteristics of the equations, and integrates forward in time using an implicit $4^{\mathbf{th}}$ order Runge-Kutta method. Among the features of our method is that it is easy to implement, efficient, and well suited for vectorization and parallelization. As is the case for all high-order methods, it can be applied to problems with shocks, but it is most effective when the underlying solution is smooth.

Our method is described in Section 2. In Section 3 we discuss the effect that the smoothness of the underlying problem has on our method, while in Section 4 we show a numerical result. In Section 5 we compare the merits of our scheme to those of other approaches, and we discuss possible improvements. Finally, in the Appendix, we discuss some important properties of the Runge-Kutta method we use.

2. Description of the method

In this section we describe the finite difference method we have used to calculate the dynamical properties of nonlinear periodic media of finite extent.[4] Application of this numerical technique to other, similar problems is discussed in Section 5. Since the emphasis here is on the numerical procedure we just show the final set of equations to be solved, referring the reader to Ref. 4 for a derivation of these equations and for a general discussion. The particular set of coupled mode equations we consider are

$$
\begin{aligned}
+i\frac{\partial \mathcal{F}_+}{\partial z} + i\frac{\partial \mathcal{F}_+}{\partial t} + \kappa\mathcal{F}_- + \Gamma|\mathcal{F}_+|^2\mathcal{F}_+ + 2\Gamma|\mathcal{F}_-|^2\mathcal{F}_+ = 0,\\
-i\frac{\partial \mathcal{F}_-}{\partial z} + i\frac{\partial \mathcal{F}_-}{\partial t} + \kappa\mathcal{F}_+ + \Gamma|\mathcal{F}_-|^2\mathcal{F}_- + 2\Gamma|\mathcal{F}_+|^2\mathcal{F}_- = 0,
\end{aligned}
\tag{1}
$$

where z and t are space- and time-like coordinates, $\mathcal{F}_\pm$ are the envelope functions of the forward and backward traveling modes which determine the electric field inside the system, and κ and Γ are linear and nonlinear coupling coefficients respectively. As illustrated in Fig. 1, the system has a length L and is driven by radiation coming in at $z = 0$. The amplitude of this radiation (in fact, that of its envelope function) is given by $A(t)$. Using this notation the boundary conditions now attain the well known form[4]

$$
\begin{aligned}
\mathcal{F}_+(0,t) &= A(t),\\
\mathcal{F}_-(L,t) &= 0.
\end{aligned}
\tag{2}
$$

In addition, we assume that the system initially contains no energy, so that [4]

$$
\mathcal{F}_\pm(z,0) = 0. \tag{3}
$$

In practice it is often useful to solve Eq. 1–3 in a rotating frame. This is discussed in more detail in Section 4.

Eqs. 1–3 comprise a well posed hyperbolic initial value problem which we now set out to solve. In doing so we calculate the envelope functions $\mathcal{F}_\pm(z,t)$, which, as discussed above, lead to the electric field inside the system. In addition, the amplitudes of the reflected and transmitted radiation, indicated schematically in Fig. 1, are given by $R(t) = \mathcal{F}_-(0,t)$ and $T(t) = \mathcal{F}_+(L,t)$, respectively.

Eqs. 1 have the important property that their characteristics[10] are straight lines. By applying the coordinate transformation

$$
\begin{aligned}
z &= \zeta - \tau,\\
t &= \zeta + \tau,
\end{aligned}
\tag{4}
$$

we obtain the equations

$$
\begin{aligned}
i\frac{\partial \mathcal{F}_+}{\partial \zeta} =& -\kappa \mathcal{F}_- - \Gamma|\mathcal{F}_+|^2\mathcal{F}_+ - 2\Gamma|\mathcal{F}_-|^2\mathcal{F}_+,\\
i\frac{\partial \mathcal{F}_-}{\partial \tau} =& -\kappa \mathcal{F}_+ - \Gamma|\mathcal{F}_-|^2\mathcal{F}_- - 2\Gamma|\mathcal{F}_+|^2\mathcal{F}_-,
\end{aligned}
\tag{5}
$$

so that the integration directions coincide with the characteristics. Eqs. 5 each look like *ordinary* differential equations (ODEs), which, however, are coupled through their right sides. As a consequence, if the envelope functions are known at a given point in space and time, say, (ζ_0, τ_0), one can apply Euler's method[11-14] to Eq. 5a to calculate $\mathcal{F}_+(\zeta_0+\delta\zeta, \tau_0)$, and to Eq. 5b to find $\mathcal{F}_-(\zeta_0, \tau_0+\delta\tau)$. As we show below, by suitably combining the results of several such integrations one can, starting at $t = 0$, find the envelope functions at all later times. To see how this can be accomplished we refer to Fig. 2 which shows the integration domain in the new coordinates ζ and τ. Note that according to Eqs. 4 the front of the system corresponds to $\zeta - \tau = 0$, whereas the back corresponds to $\zeta - \tau = L$. Further, as indicated in Fig. 2 by the dotted lines, time is constant along the straight lines for which $\zeta + \tau = C$. Eqs. 5 show that $\mathcal{F}_+$ is integrated to the right in Fig. 2, or, as expected, towards the back of the system, while $\mathcal{F}_-$ is integrated upwards, or towards the front.

We integrate Eqs. 5 using a finite difference scheme.[11-14] Such a procedure is illustrated in Fig. 2 for the case in which each of the envelope functions is sampled at $N + 1 = 5$ positions inside the system, so that at each time the state of system is described by $2N + 2 = 10$ complex numbers. Note that in an actual calculation the parameter N is usually chosen to be larger than four. Note further that the stepsizes (see Fig. 2) in the space- and time-directions are coupled now: to integrate over a fixed time, the round trip time $T_r = 2L$ (see Fig. 2), say, $2N$ timesteps are required. We thus see that the number of required integrations is proportional to N^2 so that it is very important to use a method which allows large stepsizes. We initially concentrate on an integration from $t = L/N$ to $t = 2L/N$ (see Fig. 2), after which we can of course repeat this procedure to integrate over any time interval (but in steps of L/N). In the finite difference scheme $\mathcal{F}_+$ and $\mathcal{F}_-$ are integrated along the horizontal and vertical dashed lines in Fig. 2 respectively, while the results can be combined at the intersections. We first consider Euler's method to perform these integrations, while we later discuss higher order schemes. As mentioned above, $2N + 2 = 10$ complex numbers are required to describe the state at a specified time. From among these, $\mathcal{F}_-(0,t)$ and $\mathcal{F}_+(L,t)$ give the amplitude during the next timestep of the reflected and transmitted light respectively. The other values can be used to integrate forward in time to produce $2N = 8$ complex numbers at the next timestep, while $\mathcal{F}_+(0,t)$ and $\mathcal{F}_-(L,t)$ follow directly from the boundary conditions (Eqs. 2), so that we again have $2N + 2$ complex numbers to describe the state of the system.

As is often the case, Euler's method is not very effective: It is a first order scheme which does not use all available information at the points of intersection at $t = 3L/(2N)$ and $t = 2L/N$. In accordance with the discussion above, in our search for higher order schemes we restrict ourselves to methods developed for *ordinary* differential equations. More specifically, we consider Runge-Kutta methods[11–14] only. These schemes seek to solve ODEs by matching a certain fixed number of terms in the Taylor series of the integrated function. They can be combined in a manner similar to that just described above for Euler's method to solve Eqs. 5.

Extending the integration to the use of higher order Runge-Kutta methods one is faced with several issues. The first of these is a desire for simplicity. Most ODE integration schemes can, at least in principle, easily be applied at positions well inside the system, but the boundaries can give rise to a myriad of "special cases." The application of a high order Adams' method,[12,14] for example, which makes use of values of the integrated function at several previous steps, is tedious to apply when integrating $\mathcal{F}_+$ close to the front of the system, since the necessary "previous" steps may be located outside the system. Such situations (and similar ones near the back) would have to be dealt with separately, thus giving rise to a bulky, error-prone code. We return to this matter in Section 5. This consideration leads us to seek one-step schemes[12–14] in which the use of previous values is limited to the last of these. The next issue is that, according to Fig. 2, when integrating from $t = L/N$ to $t = 2L/N$ using a one-step scheme, each integration path exactly crosses three others (at $t = L/N$, $t = 3L/(2N)$, $t = 2L/N$). This limits us to methods which just use the endpoints and the midpoint in each step. One might naively expect to be able to use the best known of these methods, the "classical" $4^{\mathbf{th}}$ order explicit Runge-Kutta scheme, which for integrating the equation $\frac{dy}{dx} = f(x, y)$ starting at (x_n, y_n) over an interval h is often written as[11–14]

$$\begin{aligned} k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + h/2, y_n + k_1/2), \\ k_3 &= hf(x_n + h/2, y_n + k_2/2), \\ k_4 &= hf(x_n + h, y_n + k_3), \\ y_{n+1} &= y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6, \end{aligned} \tag{6}$$

but we will now explain why this is not true. To see this most easily we should realize that each k_i is an estimate of $y_{n+1} - y_n$. However, while the final result for y_{n+1} is $4^{\mathbf{th}}$ order, the preliminary results are much less accurate: in fact the first is $1^{\mathbf{st}}$ order only. However, they are *biased* in such a way as to yield a $4^{\mathbf{th}}$ order result after the final summation in Eqs. 6, as the lower-order error terms cancel in this process. Now consider the situation in Fig. 2 again. Integrating in the vertical direction from $z = L/N$ one would need a suitably biased estimate for

$\mathcal{F}_+$ at $z = 0$, $t = 2L/N$ for k_4. However, due to the finite extent of the system, such an estimate is not available. Rather, the first boundary condition in Fig. 2 gives the *exact* value of $\mathcal{F}_+$. If this value is used, the lower-order error terms do *not* cancel completely when performing the final summation in Eqs. 6, resulting in an integration scheme which is at most $3^{\mathbf{rd}}$ order. Fig. 2 shows that a similar problem arises when integrating horizontally from $z = (N-1)L/N$. Although this corruption originates from the two boundaries of the system, it propagates inwards in time along the characteristics so that the final result at any position in the system is effectively less than $4^{\mathbf{th}}$ order. This is a fundamental problem which cannot be avoided with the algorithm in Eqs. 6, and most likely all other computationally-effective explicit Runge-Kutta schemes. Moreover, our numerical experiments show that the "classical" $4^{\mathbf{th}}$ order Runge-Kutta scheme performs even more poorly than the simple argument above would lead one to expect. This may be due in part to the use of low-order biased values at the midpoints at $t = 3L/(2N)$.

The discussion above leads to three requirements for our integration scheme: It should be (1) a one-step method which uses (2) unbiased estimates at (3) the endpoints and the midpoint of each integration step. *Collocation* methods[12,13] satisfy these requirements. In these, one approximates the function to be integrated locally by a polynomial which is required to satisfy the differential equation at a fixed number of (collocation) points. Since in this application we can specify the polynomial at exactly three equidistant points, the polynomial can be $2^{\mathbf{nd}}$ degree at most. As shown in the Appendix, this leads to a $4^{\mathbf{th}}$ order integration method. Notice that this approach is similar to Simpson's rule for quadrature, in which the integrated function is also approximated locally by a parabola and which is a $4^{\mathbf{th}}$ order method as well.[11] Using similar notation as above, the method we have used can be written as

$$\begin{aligned} y_{n+\frac{1}{2}} &= y_n + h\Big[a_{21} f(x_n, y_n) + a_{22} f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + a_{23} f(x_{n+1}, y_{n+1})\Big], \\ y_{n+1} &= y_n + h\Big[a_{31} f(x_n, y_n) + a_{32} f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + a_{33} f(x_{n+1}, y_{n+1})\Big], \end{aligned} \tag{7}$$

where the numerical values of the coefficients are listed in Table 1. A derivation of this formula is given in the Appendix. The application of Eqs. 7 to the problem at hand follows again immediately from Fig. 2. To integrate from $t = L/N$ to $t = 2L/N$ the algorithm is applied $2N = 8$ times. The values thus produced, in combination with the boundary conditions (Eqs. 2), completely specify the state of the system at $t = 2L/N$.

Thus far we have not discussed how the set of implicit equations (Eqs. 7) are to be solved. In principle this can be a large problem since the $2N$ applications of Eqs. 7 necessary to integrate over a single timestep are not independent (see Fig. 2), thus giving rise to a set of $4N$ coupled complex nonlinear algebraic equations. As shown below, however, we have in general good estimates for all $y_{n+\frac{1}{2}}$ and y_{n+1}. Starting with these estimates, one can solve the implicit equations in several different ways, including Newton's method and simple iteration,[12–15] the latter giving rise to a predictor-corrector scheme. Since such a scheme effectively decouples all equations, allowing them to be solved one-by-one, simple iteration is in this instance the method of choice, provided that the problem is nonstiff, which is the case here. We discuss this in more detail in the Appendix.

The implementation is as follows. (1) We first estimate values for $y_{n+\frac{1}{2}}$ and y_{n+1} for all $4N$ equations. (2) Then using these estimates we evaluate the right side of Eqs. 7 producing better estimates for $y_{n+\frac{1}{2}}$ and y_{n+1} for all $4N$ equations. (3) We repeat step (2) n_I times until the required accuracy has been reached, always using the most recently computed values for $y_{n+\frac{1}{2}}$ and y_{n+1} when evaluating the right side of Eqs. 7. Since each iteration requires function evaluations at all collocation points at $t = 3L/(2N)$ and $t = 2L/N$, it is important to select n_I carefully. It is common to use $n_I = 1$ in predictor-corrector methods. However, since in the present application the minimization of N is crucial (because the total number of integrations is proportional to N^2), we take $n_I = 3$ to enhance stability and to enable us to use smaller values of N than would be the case for $n_I = 1$ or 2. It should be stressed that this value of n_I applies to our particular case only and should be reconsidered for any other application.

As mentioned above, we need good estimates for $y_{n+\frac{1}{2}}$ and y_{n+1}. Suppose that we obtained the values of the envelope functions at $t = L/N$ after an integration starting from $t = 0$. As described in the Appendix, this involves approximating the envelope functions within this interval by parabolas. It is natural to extrapolate these parabolas to estimate the envelopes in the next time interval. This procedure gives rise to equations similar to Eqs. 7 but with different coefficients:

$$\begin{aligned} y^{(0)}_{n+\frac{3}{2}} &= y_n + h\left[\hat{a}_{21} f(x_n, y_n) + \hat{a}_{22} f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + \hat{a}_{23} f(x_{n+1}, y_{n+1})\right], \\ y^{(0)}_{n+2} &= y_n + h\left[\hat{a}_{31} f(x_n, y_n) + \hat{a}_{32} f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + \hat{a}_{33} f(x_{n+1}, y_{n+1})\right], \end{aligned} \tag{8}$$

where the $\hat{a}_{ij}$ are also listed in Table 1. For details we refer once more to the Appendix.

The extrapolation equations (Eqs. 8) yield all necessary estimates with two exceptions. The first of these occurs at the beginning of the integration where we have no estimate from a previous step. The second exception, due to the finite extent of the system, occurs at the edges. Fig. 2 shows that we have no estimates from a previous step when integrating horizontally from $z = 0$, $t = L/N$, or when integrating vertically from $z = L$, $t = L/N$. Recall however that we just need estimates at these points and even poor ones, produced for example by a low order method, such as Euler's or Heun's,[11,12,14] can be improved by the iteration procedure described above (see the Appendix also). Since this procedure is necessary only on the first step of the integration and near the edges, the additional CPU time required is proportional to N, rather than N^2.

The integration procedure is now clear. It is started in the way described in the preceding paragraph. For the initial condition in Eq. 3 describing an empty system this is particularly easy. After a sufficient number of iterations using Eqs. 7, Eqs. 8 are applied to extrapolate into the next time interval, while some values at the edges (see the paragraph above) are estimated separately. The newly obtained values are then iterated, and again the extrapolation procedure is applied. In this way one integrates forward in steps of $t = L/N$, with $A(t)$ describing the desired driving signal.

3. Additional considerations

In this section we discuss the effect of the smoothness of the underlying problem on our numerical results. Suppose the underlying solution to the problem has shocks propagated along the characteristics. Assume for definiteness that a shock runs along the characteristic $\tau = (t - z)/2 = c_1$ for some constant c_1 and that $\mathcal{F}_+$ has a jump discontinuity from one side of this line to the other. Consider the integration of $\mathcal{F}_-$ along a characteristic $\zeta = (t + z)/2 = c_2$ on a step for which the two characteristics cross at the midpoint of the integration step (see Fig. 2). Thus, when integrating $\mathcal{F}_-$ on this step, there is a discontinuity in the right side of Eq. 5b caused by the jump discontinuity in $\mathcal{F}_+$. As a result, $\mathcal{F}_-$ cannot be approximated well by a polynomial on this step. Another way of interpreting this loss of accuracy is to note that the $4^{\mathbf{th}}$ order ODE scheme used can be derived by matching the first five terms in the Taylor series expansion of the true and numerical solutions. Near a shock, though, the true solution does not have such an expansion, so the numerical scheme cannot achieve $4^{\mathbf{th}}$ order accuracy. Moreover, since Eqs. 5 are coupled, a deterioration in accuracy of either $\mathcal{F}_+$ or $\mathcal{F}_-$ is transferred to the other, resulting in a severe deterioration in accuracy of both. In numerical tests on problems with shocks, our scheme generally appeared to be $1^{\mathbf{st}}$ order accurate, in agreement with a simple error analysis. However, on problems with smooth solutions the numerical results exhibit $4^{\mathbf{th}}$ order convergence as expected.

Fortunately, from the physics of the phenomena, we expect the solution of our problem to be smooth, but mathematically this is the case only if the driving function $A(t)$ (see Eq. 2) is chosen carefully. For example, taking $A(t) = 1$ for $t > 0$ yields a discontinuity between the initial and front boundary conditions at $(z = 0,\ t = 0)$ that produces a shock which radiates out from $(z = 0,\ t = 0)$ along the characteristic $\tau = (t - z)/2 = 0$. When this problem is solved by our scheme, the resulting numerical solution appears to be $1^{\mathbf{st}}$ order accurate. On the other hand, if for $t \in [0, 1]$ we take $A(t)$ to be the unique polynomial of degree 9 that satisfies $A(0) = 0$, $A(1) = 1$ and $A^{(i)}(0) = A^{(i)}(1) = 0$ for $i = 1, \ldots, 4$, and if for $t > 1$ we take $A(t) = 1$, then the underlying solution to the problem is sufficiently smooth for numerical solutions with our method to exhibit $4^{\mathbf{th}}$ order convergence.

The problem specified by Eqs. 1-3 seems to be smooth except for the possible initial shocks generated by a "mismatch" of the initial and front boundary conditions at $(z = 0,\ t = 0)$. Moreover, on the smooth part of a problem with an initial shock, the numerical solution appears to have a $4^{\mathbf{th}}$ order local error. The distinction between smooth and nonsmooth problems and the resulting $2^{\mathbf{nd}}$ and $4^{\mathbf{th}}$ order convergence of our method may not be as important as it first seems for the numerical solution of the nonlinear coupled mode equations (Eqs. 1). The initial conditions and the driving function $A(t)$ for small t are quite artificial, and the use of this mathematical model is predicated upon the hope that the choice of these will not affect the physically important properties of the model's solution for large t. Therefore, it is our believe that the possible loss of accuracy when integrating past an initial shock does not affect the scheme's ability to generate physically useful results for large t, since the initial loss of accuracy can be viewed as an $O(h^2)$ perturbation of the solution at some small t. This interpretation is supported by the numerical results which provide seemingly accurate physical values for problems with initial shocks. In fact, we experienced no deterioration in the usefulness of the method for nonsmooth problems. Moreover, for the nonlinear coupled mode equations, the numerical results generated by our scheme are superior to those produced by other schemes, such as the split-step method discussed in more detail in Section 5.

4. Numerical Example

In this section we briefly present an example of the use of our method. To do so we must first specify the driving signal $A(t)$ (Eq. 2). As in our other investigations[4] we take this driving signal to be harmonically varying in time, except for smooth transitions when the magnitude of $A(t)$ changes. As mentioned in Section 2, we have found it convenient to solve Eqs. 1 in a rotating frame, chosen such that the incoming signal can be taken to be *real*. Thus if the incoming signal varies harmonically as $\sim e^{-i\delta t}$, we consider the envelope functions $\mathcal{G}_\pm(z,t) \equiv \mathcal{F}_\pm(z,t)e^{-i\delta t}$ which satisfy the equations

$$\begin{aligned} +i\frac{\partial \mathcal{G}_+}{\partial z} + i\frac{\partial \mathcal{G}_+}{\partial t} + \delta\mathcal{G}_+ + \kappa\mathcal{G}_- + \Gamma|\mathcal{G}_+|^2\mathcal{G}_+ + 2\Gamma|\mathcal{G}_-|^2\mathcal{G}_+ = 0, \\ -i\frac{\partial \mathcal{G}_-}{\partial z} + i\frac{\partial \mathcal{G}_-}{\partial t} + \delta\mathcal{G}_- + \kappa\mathcal{G}_+ + \Gamma|\mathcal{G}_-|^2\mathcal{G}_- + 2\Gamma|\mathcal{G}_+|^2\mathcal{G}_- = 0. \end{aligned} \tag{9}$$

We apply our method to these transformed equations.

Fig. 3 shows the resulting transmitted amplitude, $|\mathcal{F}_+(L,t)| = |\mathcal{G}_+(L,t)|$, for $t \leq 25$ for a calculation with $N = 28$, in which $\kappa = 5.0$, $\delta = 4.5$, $L = 1.0$, $\Gamma = 0.1$. In the rotating frame the incoming radiation has an amplitude $A(t) = A_0 = 2.0$. To avoid shocks this driving force is not applied instantaneously. Rather, it increases smoothly for $0 < t < 2$ from $A(0) = 0$ to $A(2) = A_0$ as described in Section 3. $A(t) = A_0$ for $t \geq 2$.

Fig. 3 shows that it takes somewhat more than a single time unit for the transmission signal to reach an appreciable value, as one time unit is required to travel the length of the structure. Moreover, the figure illustrates that after a short transient regime the system exhibits self-pulsing, but with a decreasing amplitude. After about $t = 100$ this amplitude stabilizes and the output is then truly periodic. Such behavior for these systems is quite common and is discussed in more detail in Ref. 4.

5. Discussion

In this section we analyze the merits of our scheme and compare these to those of other numerical methods. We also discuss possible improvements. First we should mention that our method, like most convergent methods in which one integrates over the characteristics, gives exact results for the *free-flowing problem* in which both $\kappa = 0$ and $\Gamma = 0$ (Eqs. 1). In this case radiation simply propagates unimpeded in both directions with unit velocity.

In addition, our method has the advantage that it is easy to implement and is quite versatile. It can be applied straightforwardly to any finite system of the general form given by Eqs. 5, provided it describes the interaction of two counter-propagating waves. This includes, for example, phase conjugation problems, but not directional couplers.[3] In the latter device one is interested in the interaction between two co-propagating modes so that both boundary conditions are specified at a single interface (*cf.* Eqs. 5), leading to an implementation which differs from ours. Apart from Eqs. 1, we have also successfully applied our method to a class of systems that exhibit transverse instability.[6] While the characteristics are now the same as those of Eqs. 1 (Fig. 2), no field is incident from either boundary (*cf.* Eqs. 2). Rather, the fields in the cavity grow through an interaction in the equations themselves, evolving to high-intensity solutions from an empty cavity. For this system the stability of our method is particularly desirable, since many commonly used explicit methods[11−14] prove unstable for large times.

We saw in Section 2 that the coupling of the stepsizes in the space and time directions leads us to select the largest possible stepsize for a particular problem. As a consequence, the number of iterations n_I has to be chosen rather large to avoid numerical instability. A possible improvement in solving the $4N$ implicit equations is to replace the simple iteration procedure altogether by a more sophisticated multidimensional root finding routine. Newton's method is often used for this purpose because of its quadratic convergence rate and since good initial guesses for the roots are available.[12,13] However, this method calls for inverting complex matrices of size $2N \times 2N$. Although the application of this method is likely to reduce N, the required manipulation of rather large matrices, albeit sparse, is expected to offset this gain. Moreover, it is highly unlikely that this root seeking routine is as easy to vectorize (see below) as our simple iteration procedure. For these reasons we have not pursued this option. However, if the problem were stiff (which is not the case here), then very substantial increases in the stepsize, which might be obtained by using a Newton iteration rather than our simple predictor-corrector technique, could make the former approach much more effective than the latter. This is also discussed in the Appendix.

Another possible improvement is the use of methods having order greater than four. Although we argued in Section 2 that no such schemes can be used, this argument was based upon integration with a one-step method over a time interval of length L/N. By integrating over larger intervals, for example $2L/N$ (see Fig. 2), the integrated function can be specified at five points, thus allowing for interpolation by a quartic polynomial, leading to a $6^{\mathbf{th}}$ order method. Such a scheme could be realized as either a one-step or two-step method. We implemented the one-step scheme but found it to be unsatisfactory, largely (we believe) due to the extrapolation procedure. First, the extrapolation is now very cumbersome as the number of exceptions arising from the edges is much larger than before (see the paragraph following Eqs. 8), giving rise to a very bulky, error-prone code. Second, a more fundamental problem is related to the magnitude of the coefficients for the extrapolation equations corresponding to Eqs. 8. Table 1 shows that the extrapolation coefficients $\{\hat{a}_{ij}\}$ for Eqs. 8 tend to increase with i and their signs (in the bottom row at least) alternate with j, with the sum of the entries in any row of the table remaining small. This trend increases dramatically with the order of the scheme (see the Appendix). For quartic extrapolation, the coefficients in all rows but the first alternate in sign and the bottom row has entries of magnitude 100, giving rise to potential instability when integrating with large stepsizes. As a consequence, we have not succeeded in using the one-step $6^{\mathbf{th}}$ order scheme fruitfully. We believe the two-step scheme would be equally unsatisfactory.

Another important advantage is the possibility to vectorize or to parallelize the procedure. In particular, the extrapolation and each iteration necessary to integrate over a single timestep must be performed an equal number of times for all samples of the two envelope functions. Since Eqs. 7 and 8 apply to every one of these samples in an identical way, the numerical procedure can be vectorized to a vector length N. We have demonstrated viability of this option by running our code on a CRAY X-MP, where a program in which $N = 60$ ran at well over 90 mflops. Similarly, the method could be parallelized effectively in a straightforward manner.

A number of other methods could have been applied to the equations we have considered here (Eqs. 1). For comparison we have tested our scheme against the split-step method,[7] which is widely used in nonlinear optics. We follow the prescription of Agrawal[7] and divide the equations into two parts, each to be integrated separately and then combined in alternate steps. Usually one divides the problem into linear and nonlinear parts, which are integrated in Fourier space and real space respectively. Here we include the linear coupling terms (the third terms in Eqs. 1) in the real space integrations, so that the remaining integration becomes just a free propagation along a characteristic (and Fourier transformations can be avoided). The resulting algorithm is $2^{\mathbf{nd}}$ order in time.[7] Since our method is $4^{\mathbf{th}}$ order, it will be more efficient if very accurate results are required. However, even for low accuracy of $\sim 0.1\%$ our method requires only half as many grid points as the split step method, and is found to be substantially faster.

In conclusion, we have presented a finite difference method to integrate a hyperbolic set of two partial differential equations. By using the characteristics, each of the equations contains a partial derivative with respect to a single independent variable only, thus allowing us to apply methods developed for ODEs. The choice of an implicit $4^{\mathbf{th}}$ order Runge-Kutta scheme leads to a versatile method which is easy to vectorize or parallelize.

6. Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, the Information Technology Research Centre of Ontario, and the Ontario Laser and Lightwave Research Centre. The authors are grateful to the Ontario Centre for Large Scale Computation for providing the opportunity for a trial run of their code on its CRAY X-MP machine.

Appendix

In this section, we review one-step collocation methods for initial value problems (IVPs) for ordinary differential equations (ODEs) of the form

$$\begin{aligned} &y'(x) = f(x, y(x)) \quad \text{for} \quad x \in [a, b], \qquad y : \mathcal{R} \to \mathcal{C}^m \qquad f : \mathcal{R} \times \mathcal{C}^m \to \mathcal{C}^m, \\ &y(a) = y_0. \end{aligned} \tag{10}$$

We pay particular attention to predictor-corrector implementations of these schemes, such as the one outlined in Section 2. Many of the basic results summarized here are discussed in more detail in Ref. 12 (particularly pages 206–209) and Ref. 13 as well as other texts on the numerical solution of IVPs for ODEs.

As noted in Section 2, collocation schemes are based on an underlying polynomial approximation $u_n(x)$ to $y(x)$ on each interval $[x_n, x_{n+1}]$ of the numerical integration. Since $y(x) \in \mathcal{C}^m$, we must take $u_n(x) \in \mathcal{C}^m$ also, with each of the m components of the vector $u_n(x)$ being a normally different polynomial of degree s. The collocation scheme is determined by the choice of the integer s and the s real distinct *collocation points*, $c_1, \ldots, c_s$, usually chosen to be in $[0, 1]$. Once these are given, $u_n(x)$ is uniquely determined by

$$\begin{aligned} u_n'(x_n + c_i h) &= f(x_n + c_i h, u_n(x_n + c_i h)), \qquad i = 1, \ldots, s, \\ u_n(x_n) &= y_n, \end{aligned} \tag{11}$$

provided f is sufficiently smooth and h is small enough. The first s conditions in Eqs. 11, referred to as the *collocation conditions*, force $u_n(x)$ to satisfy the ODE in Eqs. 10 at the s collocation points in the hope that this will lead to a good approximation $u_n(x)$ to $y(x)$ on $[x_n, x_{n+1}]$. Then y_{n+1} is taken to be $u_n(x_{n+1})$, and the scheme proceeds to the next step from x_{n+1} to $x_{n+2} = x_{n+1} + h$ by defining another polynomial $u_{n+1}(x)$ over $[x_{n+1}, x_{n+2}]$ in a similar manner. The numerical integration, of course, starts with $n = 0$, $x_0 = a$ and $y_0 = y(a)$.

In Section 2, we used a fixed stepsize h throughout the integration for simplicity. This restriction arises from the PDE application, not the ODE method itself; we could just as easily have used a stepsize h_n in Eqs. 11 together with some strategy for varying h_n, as discussed in Ref. 12 for example.

Although the representation given in Eqs. 11 is very useful, it does not in itself lead to an efficient implementation of a collocation scheme. To this end, it is very instructive to observe that one-step collocation schemes form a subclass of implicit Runge-Kutta methods. To see this, we introduce the as yet undetermined values $y_{n+c_i} = u_n(x_n + c_i h)$, $i = 1, \ldots, s$. Assuming these values exist, which we establish below, polynomial interpolation theory[14] ensures that $u'_n(x)$ is the unique polynomial of degree $s-1$ that satisfies the s collocation conditions in Eqs. 11. Moreover, $u'_n(x)$ may be represented in Lagrange form[14] as

$$u'_n(x) = \sum_{i=1}^{s} l_i(x) f(x_n + c_i h, y_{n+c_i}),$$

where

$$l_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{s} \frac{x - (x_n + c_j h)}{h(c_i - c_j)} \tag{12}$$

are the *elementary Lagrange interpolatory polynomials* of degree $s-1$ associated with the interpolation points $\{x_n + c_i h\}_{i=1}^{s}$. Using this representation, we can rewrite Eqs. 11 in integral form as

$$u_n(x) = u_n(x_n) + \int_{x_n}^{x} u_n'(\hat{x})\,d\hat{x} = y_n + \sum_{j=1}^{s} \left(\int_{x_n}^{x} l_j(\hat{x})\,d\hat{x} \right) f(x_n + c_j h, y_{n+c_j}) \quad (13)$$

whence taking $x = x_n + c_i h$ we get

$$y_{n+c_i} = u_n(x_n + c_i h) = y_n + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, y_{n+c_j}) \quad (14)$$

where the fixed coefficients $\{a_{ij}\}$ satisfy

$$a_{ij} = \frac{1}{h} \int_{x_n}^{x_n + c_i h} l_j(\hat{x})\,d\hat{x} = \int_0^{c_i} \left(\prod_{\substack{k=1 \\ k \neq j}}^{s} \frac{\tilde{x} - c_k}{c_j - c_k} \right) d\tilde{x}. \quad (15)$$

Similarly,

$$y_{n+1} = u_n(x_n + h) = y_n + h \sum_{j=1}^{s} b_j f(x_n + c_j h, y_{n+c_j}) \quad (16)$$

where each b_j satisfies equations similar to Eqs. 15 but with c_i replaced by unity.

Since $c_1, \ldots, c_s$ are distinct, Eqs. 14 contain s equations in the s unknowns $\{y_{n+c_i}\}$. Although these equations are implicit if at least one c_i is nonzero, the Contraction Mapping Theorem[14,15] ensures that they have a unique solution provided f is sufficiently smooth and h is small enough. Moreover, this solution can be found by simple iteration, which corresponds to a predictor-corrector implementation of the collocation scheme, as discussed in more detail at the end of this section.

It follows immediately from Eqs. 15 that if $c_i = 0$, then the associated $a_{ij} = 0$ for $j = 1, \ldots, s$ and the corresponding equation for y_{n+c_i} in Eqs. 14 reduces to the identity $y_n = y_n$. Similarly, if $c_i = 1$, then the associated $a_{ij} = b_j$ for $j = 1, \ldots, s$ and the corresponding equations for y_{n+1} given by Eqs. 14 and 16 are identical.

For the 3-stage collocation scheme discussed in Section 2, $c_1 = 0$, $c_2 = \frac{1}{2}$ and $c_3 = 1$. Consequently, we need to solve Eqs. 7 for $y_{n+\frac{1}{2}}$ and y_{n+1} only, since the third y-value, y_n, needed for the scheme is known at the start of the step. The coefficients $\{a_{ij}\}$ in Eqs. 7 are given by Eqs. 15; their actual numerical values are listed in Table 1.

The more traditional form of a Runge-Kutta method can be obtained from Eqs. 14 and 16 by letting $k_i = hf(x_n + c_i h, y_{n+c_i})$ and substituting the value for y_{n+c_i} from Eqs. 14 into this last equation, to get

$$k_i = hf(x_n + c_i h,\; y_n + h\sum_{j=1}^{s} a_{ij} f(x_n + c_j h, y_{n+c_j}))$$

and again using $k_j = hf(x_n + c_j h, y_{n+c_j})$ to yield finally

$$\begin{aligned} k_i &= hf(x_n + c_i h, y_n + \sum_{j=1}^{s} a_{ij} k_j), \quad i = 1, \ldots, s, \\ y_{n+1} &= y_n + \sum_{j=1}^{s} b_j k_j. \end{aligned}$$

However, the formulation given by Eqs. 14 and 16 is more useful for our application.

It is instructive to note that each collocation scheme is closely related to an interpolatory quadrature rule, by which we mean a quadrature rule for which the integrand $g(x)$ is approximated by an interpolating polynomial $p(x)$, and $\int p(x)\,dx$ is used to approximate $\int g(x)\,dx$. To see this, observe that, if $f(x,y)$ does not depend on y, then the collocation (i. e., interpolation) conditions in Eqs. 11 reduce to $u_n'(x_n + c_i h) = f(x_n + c_i h)$ for $i = 1, \ldots, s$, which uniquely determine the polynomial $u_n'(x)$ of degree $s-1$. Therefore, we can use Eqs. 16 directly, without having to compute the auxiliary values $\{y_{n+c_i}\}$ from Eqs. 14, to yield

$$y_{n+1} - y_n = h\sum_{j=1}^{s} b_j f(x_n + c_j h) = \int_{x_n}^{x_{n+1}} u_n'(x)\,dx \approx \int_{x_n}^{x_{n+1}} f(x)\,dx. \tag{17}$$

The coefficients $\{c_i\}$ and $\{b_i\}$ are the nodes and weights, respectively, of the related interpolatory quadrature rule. The collocation formula in Eqs. 7 is related in this way to Simpson's rule,[11,14] an interpolatory quadrature rule. Both formulas are based on interpolating polynomials of degree 2 (i.e., parabolas).

To determine the order of a collocation formula and to understand how the predictor given by Eqs. 8 works, we introduce the *local solution* $y_n(x)$ to the *local* IVP

$$y_n'(x) = f(x, y_n(x)), \qquad y_n(x_n) = y_n. \tag{18}$$

Note that $y_n(x)$ satisfies the same ODE as $y(x)$ but a different initial condition in general. Now observe that $u_n(x)$ satisfies a perturbed version of the local IVP 18:

$$u_n'(x) = f(x, u_n(x)) + [u_n'(x) - f(x, u_n(x))], \qquad u_n(x_n) = y_n, \tag{19}$$

where the perturbation satisfies $[u_n'(x) - f(x, u_n(x))] = O(h^s)$. This follows from the collocation conditions in Eqs. 11 and from the error formula for polynomial interpolation.[14] Based on this, one can show that

$$y_n(x) - u_n(x) = O(h^{s+1}) \quad \text{for} \quad x \in [x_n, x_{n+2}].$$

See Theorem 7.9 of Ref. 12 for details. Note that we have extended the result there slightly by taking $x \in [x_n, x_{n+2}]$ rather than the basic interval $[x_n, x_{n+1}]$ over which the step is taken. This allows us to use the $O(h^{s+1})$ error bound when extrapolating into the neighbouring interval $[x_{n+1}, x_{n+2}]$ as well as on the basic interval $[x_n, x_{n+1}]$.

It can happen, however, that the *local error* $y_n(x_{n+1}) - u_n(x_{n+1}) = O(h^{p+1})$ for $p > s$. To see how this can arise, note that the perturbation $[u_n'(x) - f(x, u_n(x))]$ in Eqs. 19 is zero at $x_n + c_i h$ for $i = 1, \ldots, s$ because of the collocation conditions in Eqs. 11. Using this observation and standard perturbation analysis for IVPs, one can show that the order p of the collocation formula is the same as that of its associated quadrature formula given in Eqs. 17. It then follows from quadrature theory that the order p is the largest integer such that this quadrature rule is exact for all polynomials of degree $p-1$. Again see Theorem 7.8 of Ref. 12 for details. For the 3-stage collocation scheme discussed in Section 2, $p = 4$ since the associated quadrature formula, Simpson's rule, is exact for cubics.[11,14]

Since Runge-Kutta formulas are always ($h \to 0$) stable, the fundamental "Stability + Consistency $\Rightarrow$ Convergence" theorem[12,14] ensures that the collocation scheme is convergent of order $O(h^p)$. That is, the *global error* at any gridpoint x_n satisfies $y(x_n) - y_n = O(h^p)$. (Note the distinction between these two errors: the local error is measured relative to the local solution $y_n(x)$ of the local IVP 18, while the global error is measured relative to the solution $y(x)$ of the IVP 10.)

An advantage of collocation schemes over general implicit Runge-Kutta formulas is that, once the $\{y_{n+c_i}\}$ have been computed for step n, they determine by Eqs. 13 the associated polynomial $u_n(x)$, which can be used locally as an approximation to either $y_n(x)$ or $y(x)$. One important application of this is to predict accurate estimates $y^{(0)}_{n+1+c_i} = u_n(x_{n+1} + c_i h)$ of $y_{n+1+c_i} = u_{n+1}(x_{n+1} + c_i h)$ for the next step by extrapolation. More specifically, Eqs. 13 yield

$$y^{(0)}_{n+1+c_i} = u_n(x_{n+1} + c_i h) = y_n + h \sum_{i=1}^{s} \hat{a}_{ij} f(x_n + c_i h, y_{n+c_i}) \tag{20}$$

where the fixed coefficients $\{\hat{a}_{ij}\}$ satisfy

$$\hat{a}_{ij} = \frac{1}{h} \int_{x_n}^{x_{n+1}+c_i h} l_j(\hat{x})\, d\hat{x} = \int_0^{1+c_i} \left(\prod_{\substack{k=1 \\ k \neq j}}^{s} \frac{\tilde{x} - c_k}{c_j - c_k} \right) d\tilde{x}. \tag{21}$$

Note that if $c_i = 0$, then the associated $\hat{a}_{ij} = b_j$ for $j = 1, \ldots, s$ and consequently the corresponding $y^{(0)}_{n+1} = y_{n+1}$.

For the 3-stage collocation scheme discussed in Section 2, $c_1 = 0$, $c_2 = \frac{1}{2}$ and $c_3 = 1$. Consequently, the predicted value $y^{(0)}_{n+1} = y_{n+1}$ for $c_1 = 0$ comes for free from the previous step. The coefficients $\{\hat{a}_{ij}\}$ in Eqs. 8 for $i = 2$ and 3 are given by Eqs. 21; their actual numerical values are listed in Table 1.

To determine the accuracy of the $\{y^{(0)}_{n+1+c_i}\}$, we first introduce $y_{n+1}(x)$, the solution of a local IVP similar to 18 but with the initial condition replaced by $y_{n+1}(x_{n+1}) = y_{n+1}$. An application of a standard perturbation result for IVPs[12] shows that $y_n(x) - y_{n+1}(x) = O(h^{p+1})$ for $x \in [x_{n+1}, x_{n+2}]$, since $y_n(x)$ and $y_{n+1}(x)$ satisfy the same ODE and

$$y_n(x_{n+1}) - y_{n+1}(x_{n+1}) = y_n(x_{n+1}) - y_{n+1} = O(h^{p+1})$$

is the local error for the formula on step n. We showed above that $y_n(x) - u_n(x) = O(h^{s+1})$ for $x \in [x_n, x_{n+2}]$, and it follows easily that a similar result, $y_{n+1}(x) - u_{n+1}(x) = O(h^{s+1})$ for $x \in [x_{n+1}, x_{n+3}]$, holds on the next step. Therefore, we have for $i = 1, \ldots, s$

$$\begin{aligned}
y^{(0)}_{n+1+c_i} - y_{n+1+c_i} &= u_n(x_{n+1} + c_i h) - u_{n+1}(x_{n+1} + c_i h) \\
&= [u_n(x_{n+1} + c_i h) - y_n(x_{n+1} + c_i h)] \\
&\qquad + [y_n(x_{n+1} + c_i h) - y_{n+1}(x_{n+1} + c_i h)] \\
&\qquad + [y_{n+1}(x_{n+1} + c_i h) - u_{n+1}(x_{n+1} + c_i h)] \\
&= O(h^{s+1})
\end{aligned}$$

(We implicitly assumed above for simplicity that $0 \leq c_i \leq 1$, but this argument can be generalized easily to show $y^{(0)}_{n+1+c_i} - y_{n+1+c_i} = O(h^{s+1})$ for any fixed set of real distinct $c_1, \ldots, c_s$.)

As noted in the discussion following Eqs. 14, we use simple iteration to solve for the y_{n+c_i} on step n, giving rise to the corrector iteration

$$y^{(k)}_{n+c_i} = y_n + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, y^{(k-1)}_{n+c_j}) \tag{22}$$

for $k = 1, \ldots, n_I$. If no $c_i = 1$, then y_{n+1} is calculated from Eqs. 16, but with $y_{n+c_i}^{(n_I-1)}$ in place of the y_{n+c_i}. For the formula in Section 2, however, $c_3 = 1$, so $y_{n+1}^{(k)}$ is automatically calculated by Eq. 22, and we take $y_{n+1} = y_{n+1}^{(n_I)}$. The predictor for the next step uses either $y_{n+c_i}^{(n_I-1)}$ or $y_{n+c_i}^{(n_I)}$ in place of y_{n+c_i} in Eqs. 20. In the former case, the associated f's have already been calculated and no further function evaluations are required. However, for "large h", the use of the $y_{n+c_i}^{(n_I)}$ may lead to a more reliable predictor. There are many variants of this sort of the basic predictor-corrector scheme described here.[12]

To gain some insight into how many iterations n_I of Eq. 22 should be performed, assume $y_{n+c_i}^{(0)} = y_{n+c_i} + O(h^{q+1})$. For the predictor discussed above, $q = s$. However, as noted in Section 2, it cannot be used on the first step of the integration or at the front or back of the system. In these cases, $q = 1$ or 2. Whether the predictor in Eqs. 20 or a lower order scheme is used, the following holds. If in a neighbourhood of the numerical solution $f(x, y)$ is Lipschitz continuous in y with Lipschitz constant L, then, by subtracting Eqs. 22 from 14 and using induction on k, we find that

$$\begin{aligned} \|y_{n+c_i} - y_{n+c_i}^{(k)}\| &= h \sum_{j=1}^{s} |a_{ij}| \, \|f(x_n + c_j h, y_{n+c_j}) - f(x_n + c_j h, y_{n+c_j}^{(k-1)})\| \\ &\leq hL \sum_{j=1}^{s} |a_{ij}| \, \|y_{n+c_j} - y_{n+c_j}^{(k-1)})\| = O(h^{q+k+1}). \end{aligned} \tag{23}$$

Similarly,

$$\|y_{n+1} - y_{n+1}^{(k)}\| = O(h^{q+k+1}).$$

Therefore, $k = p - q$ corrections suffice to attain $y_n(x_{n+1}) - y_{n+1}^{(k)} = O(h^{p+1})$, and thereby an order p predictor-corrector implementation of the collocation scheme. Of course, if $p = q$, at least 1 iteration should be performed.

Eqs. 23 show that the predictor-corrector iteration converges linearly if

$$hL \sum_{j=1}^{s} |a_{ij}| < 1 \quad \text{for} \quad i = 1, \ldots, s \tag{24}$$

where L is the Lipschitz constant, and, moreover, the convergence factor is proportional to h. It is easy to find examples for which the iteration diverges if inequality 24 is not satisfied. Furthermore, this inequality suggests why it may be advantageous to take $n_I > p - s$ for "large h", as recommended in Section 2. The predictor is often quite poor and the convergence factor near 1 in such cases. Consequently, a few additional corrections help to stabilize the predictor-corrector scheme and to improve its accuracy, although the formal order of the method is not increased. Some initial experimentation with the choice of n_I and variants of the predictor-corrector scheme is advised to tune the method to the problem at hand.

A predictor-corrector implementation of a formula is effective only if satisfying inequality 24 does not impose too severe a restriction on the *natural stepsize* for the problem. As noted in Section 2, predictor-corrector implementations are effective for nonstiff IVPs but not for stiff ones. For the latter class, some variant of Newton's method[13,15] should be used instead to solve the implicit Eqs. 14.

Intuitively, a stiff problem is one which is *super stable* in the sense that, over a significant portion of the interval of integration, the solution to be computed is slowly varying and all other nearby trajectories converge rapidly to it. One useful, but not completely rigorous, way of characterizing this mathematically is that, over a significant portion of the interval of integration, the Jacobian $f_{\boldsymbol{y}}(t, y(t))$ of the IVP 10 has (1) some eigenvalue λ_i for which $\Re(\lambda_i)\,|b-a| \ll -1$, (2) no eigenvalue λ_i for which $\Re(\lambda_i)\,|b-a| \gg 1$, and (3) no eigenvalue λ_i for which $|\Im(\lambda_i)|\,|b-a| \gg 1$ unless also $\Re(\lambda_i)\,|b-a| \ll -1$. As a result, the natural stepsize h for the smooth part of a stiff problem is such that $h|\lambda_i| \gg 1$ for the eigenvalues λ_i satisfying (1). However, if we define L to be the Lipschitz constant for $f_{\boldsymbol{y}}(x, y)$ in a neighborhood of that section of the solution, then, since for any eigenvector v_i associated with an eigenvalue λ_i of $f_{\boldsymbol{y}}(x, y)$ and any nonzero scalar ϵ, $L\|v\| \geq \|f(x, y + \epsilon v) -$

$f(x,y)\|/\epsilon \to \|f_y(x,y)v\| = |\lambda_i|\,\|v\|$ as $\epsilon \to 0$, we also have $hL \gg 1$. We thus see from inequality 23 that if the natural step size h is taken, the predictor-corrector scheme may not converge. Rather, a much smaller step size may have to be used. A nonstiff problem is one which is not super stable in the sense described above, and typically the natural stepsize for the problem is such that $hL < 1$. We refer the interested reader to Section 1.1 of Ref. 13 for an excellent, more extensive, discussion of stiffness.

Finally, we note that collocation schemes always have

$$\sum_{j=1}^{s} b_j = 1, \qquad \sum_{j=1}^{s} a_{ij} = c_i, \qquad \sum_{j=1}^{s} \hat{a}_{ij} = 1 + c_i \qquad \text{for} \quad i = 1, \ldots, s,$$

because $\sum_{j=1}^{s} l_j(x) = 1$ for all x due to the uniqueness of polynomial interpolation,[14] where the $\{l_j(x)\}$ are the elementary Lagrange interpolatory polynomials given by Eq. 12. For some formulas, though, such as the one in Section 2, some b_j's, a_{ij}'s or $\hat{a}_{ij}$'s may be negative. Consequently, $\sum_{j=1}^{s} |b_j|$, $\sum_{j=1}^{s} |a_{ij}|$ or $\sum_{j=1}^{s} |\hat{a}_{ij}|$ may be larger than 1, $|c_i|$ or $|1 + c_i|$, respectively. As noted in Section 3, this happens more often for the $\hat{a}_{ij}$'s than for either the b_j's or a_{ij}'s, and, moreover, the likelihood of it occurring increases with s. The reason for this is that we generally choose all $c_i \in [0,1]$, in which case all roots of each $l_j(x)$ (Eq. 12) are in $[x_n, x_{n+1}]$, and consequently $l_j(x)$ alternates in sign for $x \in [x_n, x_{n+1}]$, while it is of constant sign for $x \in [x_{n+1}, x_{n+2}]$, with the sign alternating with j. Moreover, $|l_j(x)|$ tends to be much larger on $[x_{n+1}, x_{n+2}]$ than on $[x_n, x_{n+1}]$. Furthermore, $|l_j(x)|$ tends to increase with s, particularly for $x \in [x_{n+1}, x_{n+2}]$. The values for the coefficients listed in Table 1 for the $4^{\mathbf{th}}$ order scheme discussed in Section 2 and our comments in Section 3 concerning the $6^{\mathbf{th}}$ order scheme bear this out. As noted in Section 3, this undesirable characteristic can lead to poor performance of a scheme — the predictor in particular — for "large h".

Table 1. Numerical values of the coefficients used in the integration and extrapolation procedures in Eqs. 7 and 8. Note that, as required by these equations, the entries in each row add up to $\frac{1}{2}$,1, $\frac{3}{2}$, 2, respectively.

$a_{21} = 5/24$	$a_{22} = 1/3$	$a_{23} = -1/24$
$a_{31} = 1/6$	$a_{32} = 2/3$	$a_{33} = 1/6$
$\hat{a}_{21} = 3/8$	$\hat{a}_{22} = 0$	$\hat{a}_{23} = 9/8$
$\hat{a}_{31} = 4/3$	$\hat{a}_{32} = -8/3$	$\hat{a}_{33} = 10/3$

References

1. A. Yariv, "Coupled mode theory for guided wave optics," IEEE J. Quantum Electron. **QE-9**, 919-933 (1973).

2. D. Marcuse, *Theory of dielectric optical waveguides*, (Academic, New York, 1974).

3. S. M. Jensen, "The nonlinear coherent coupler," IEEE J. Quantum Electron. **QE-9**, 1580-1583 (1982).

4. C. Martijn de Sterke and J. E. Sipe, "Switching behavior of finite periodic nonlinear media," Phys. Rev. A **42**, 2558 (1990).

5. H. G. Winful and G. D. Cooperman, "Self-pulsing and chaos in distributed feedback bistable optical devices," Appl. Phys. Lett. **40**, 298-300 (1982).

6. B. D. Robert and J. E. Sipe, "Instability in an illuminated nonlinear waveguide: a phase conjugation effect," Opt. Lett. **15**, 261 (1990).

7. G. P. Agrawal, *Nonlinear fiber optics*, (Academic, San Diego, 1989), Ch. 7.

8. For a recent overview, see e.g. J. Opt. Soc. Am. B **5**, no. 2 (1988).

9. H. G. Winful, J. H. Marburger, and E. Garmire, "Theory of bistability in nonlinear distributed feedback structures," Appl. Phys. Lett. **35**, 379-381 (1979).

10. N. Bleistein, *Mathematical methods for wave phenomena*, (Academic, Orlando, 1984).

11. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, (Cambridge, 1986).

12. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations* I, (Springer, Berlin, 1987).

13. K. Dekker and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, (North-Holland, Amsterdam, 1984).

14. E. Isaacson and H. B. Keller, *Analysis of numerical methods*, (Wiley, New York, 1966).

15. J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, (Academic Press, New York, 1970).

Captions

1. Schematic of our geometry. The envelope of the incoming radiation is denoted by $A(t)$, while the reflected and transmitted radiation are designated by $R(t)$ and $T(t)$ respectively.

2. Integration domain in the coordinates ζ and τ. The boundaries of this domain are given by thick solid lines. The short dashed lines indicate points of equal time. In a finite difference scheme the integration takes place along the long dashed lines. Note that in the figure the envelopes are sampled at $N+1=5$ positions only inside the system. In our calculations N is taken to be much larger. The parameter T_r designates a single round trip time.

3. Numerical results for a calculation in which our method is applied to Eqs. 9. Shown is the magnitude of the transmitted radiation $|\mathcal{F}(L,t)|$ as a function of time. Eventually (after about $t = 100$) the amplitude of the signal settles, and the output is then truly periodic. The parameters are $\kappa = 5.0$, $\delta = 4.5$, $\Gamma = 0.1$, $L = 1.0$, and, in the rotating frame, a constant incoming signal of $A(t) = A_0 = 2.0$. As discussed in Section 3 this driving signal is taken to increase smoothly from $A(0) = 0$ to $A(2) = A_0$ to avoid shocks. $A(t) = A_0$ for $t \geq 2$.