

oml

**OAK RIDGE
NATIONAL
LABORATORY**

MARTIN MARIETTA

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES



3 4456 0329403 0 ORNL/TM-11316

A Comparison Between the Propagators Method and the Decomposition Method for Nonlinear Equations

Y. Y. Azmy
D. G. Cacuci
V. Protopopescu

OAK RIDGE NATIONAL LABORATORY

CENTRAL RESEARCH LIBRARY

CIRCULATION SECTION

ROOM ROOM 173

LIBRARY LOAN COPY

DO NOT TRANSFER TO ANOTHER PERSON

If you wish someone else to see this
report, send to them with report and
the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831, prices available from (615) 576-8401. FTS 526-8401

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.
NTIS price codes—Printed Copy: A03 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Engineering Physics and Mathematics

**A COMPARISON BETWEEN THE PROPAGATORS
METHOD AND THE DECOMPOSITION
METHOD FOR NONLINEAR EQUATIONS**

Y. Y. Azmy, D. G. Cacuci,* and V. Protopopescu

*Department of Chemical and Nuclear Engineering
3323 Engineering II
University of California at Santa Barbara
Santa Barbara, California 93106

DATE PUBLISHED —JANUARY 1990

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES



3 4456 0329403 0

CONTENTS

ABSTRACT	v
1. INTRODUCTION	1
2. PROPAGATORS FOR NONLINEAR SYSTEMS: A BRIEF THEORETICAL REVIEW	2
3. APPLICATION OF THE NONLINEAR PROPAGATOR METHOD TO THE RICCATI EQUATION	4
4. APPLICATION OF THE DECOMPOSITION METHOD TO THE RICCATI EQUATION	8
5. NUMERICAL RESULTS AND CONCLUSIONS	10
REFERENCES	17

ABSTRACT

Recently, a new formalism for solving nonlinear problems has been formulated. The formalism is based on the construction of advanced and retarded propagators that generalize the customary Green's functions in linear theory. One of the main advantages of this formalism is the possibility of transforming nonlinear differential equations into nonlinear integral equations that are usually easier to handle theoretically and computationally. The aim of this paper is to compare, on an example, the performances of the propagator method with other methods used for nonlinear equations, in particular, the decomposition method. The propagator method is stable, accurate, and efficient for all initial values and time intervals considered, while the decomposition method is unstable at large time intervals, even for very conveniently chosen initial conditions.

1. INTRODUCTION

Recently, a new formalism for solving nonlinear problems has been formulated by Cacuci, Perez, and Protopopescu [3,4]—for scalar problems—and subsequently, generalized by Cacuci and Protopopescu [5] to multicomponent (i.e., matrix) systems. As shown in Refs. [3–5], this formalism is the natural generalization to nonlinear problems of the Green’s function formalism in linear theory, and is canonically and exactly applicable to any nonlinear operator equation when the Gâteaux-derivatives of the respective operators exist. Fundamental to this formalism is the construction of advanced (forward) and retarded (backward) propagators; these propagators generalize the customary Green’s functions, to which they reduce exactly for linear problems. In particular, this formalism does not contain any inherent approximations, and is free of linearization of “smallness” assumptions.

On the other hand, the decomposition method proposed by Adomian [1] is also presented as an exact method for solving nonlinear equations—involving no linearization or “smallness” assumptions. This method is based on the decomposition of the nonlinear operator under consideration into an easily invertible linear operator and a polynomial expansion—an infinite series, in general—of the remaining nonlinearities.

The aim of this work is to present a comparison between the nonlinear propagators and decomposition methods for a simple, but often encountered, Riccati equation whose analytical solution can be obtained by standard methods. Following a brief review, presented in Section 2 of the nonlinear propagator formalism originally developed in Refs. [3–5], the application of this formalism to the Riccati equation is presented in Section 3. The decomposition method is applied to the Riccati equation in Section 4. The numerical results are presented in Section 5; these results allow a comparison of the stability (or lack thereof), accuracy, and computational efficiency properties of these two methods; this comparison also includes results obtained by two standard, often-used methods (finite difference and Picard iteration) for solving nonlinear equations.

2. PROPAGATORS FOR NONLINEAR SYSTEMS: A BRIEF THEORETICAL REVIEW

For the reader's convenience, a brief review of the formalism developed in Refs. [3–5] will be specifically applied to a scalar (one component) equation, in accordance with the application considered in this work. The general nonlinear equation in abstract form reads

$$N(u) + \delta \cdot \Gamma(u) = f + \delta \cdot g, \quad (2.1)$$

where $N(u)$ represents the nonlinear equation itself, $\Gamma(u)$ represents the nonlinear initial/boundary conditions, f represents the volume source, and g represents the boundary source (including initial conditions). The source term f includes the inhomogeneities of the nonlinear operators, so we can consider without loss of generality that $N(0) = 0$, and $\Gamma(0) = 0$. The δ -distributions multiplying the boundary terms in Eq. (2.1) allow a formally unified abstract treatment of both boundary conditions and operators.

The vector u is considered to be an element in the Hilbert space $L_2(\Omega)$ endowed with an inner product denoted by \langle, \rangle ; throughout this work, Ω denotes the set (including the time-domain for time-dependent problems) that defines the phase space for Eq. (2.1). Since we include the boundaries in the formal treatment, Ω is a closed set containing the boundaries of the phase space underlying the problem.

The first Gâteaux-derivatives of the operators appearing on the left side of Eq. (2.1), defined as

$$[N'(u) + \delta \cdot \Gamma'(u)]h \equiv \{d/d\epsilon[N(u + \epsilon h) + \delta\Gamma(u + \epsilon h)]\}_{\epsilon=0} \quad (2.2)$$

are required to exist. The operators $N'(u), \Gamma'(u)$ depend nonlinearly on u but act linearly on the vector h .

The operator adjoint to $N'(u) + \delta \cdot \Gamma'(u)$ is defined via the usual linear duality:

$$\langle [N'(u) + \delta \cdot \Gamma'(u)]h, v \rangle = \langle h, [N'^*(u) + \delta^* \cdot \Gamma'^*(u)]v \rangle. \quad (2.3)$$

In Eq. (2.3), $N'^*(u)$ is the *formal* adjoint of $N'(u)$, and $\Gamma'^*(u)$ includes all surface operators on v . Note in Eq. (2.3), that the operator δ^* is not the same δ -distribution as in Eqs. (2.1) or (2.2), but is a distribution associated with the adjoint boundary space. This distinction is highlighted by using the symbolic notation δ^* .

Following Refs. [3–5], we define the operators

$$L(u)h \equiv \int_0^1 N'(\epsilon u)h d\epsilon, \quad \gamma(u)h \equiv \int_0^1 \Gamma'(\epsilon u)h d\epsilon, \quad (2.4)$$

and

$$L^*(u)v \equiv \int_0^1 [N'(\epsilon u)]^*v d\epsilon, \quad \gamma^*(u)v \equiv \int_0^1 [\Gamma'(\epsilon u)]^*v d\epsilon; \quad (2.5)$$

note that the operators L, γ, L^* , and γ^* still act on h and v , respectively, while retaining a nonlinear parametric dependence on u . Note also the important relationship satisfied by $L(u)$ and $\gamma(u)$:

$$[L(u) + \delta \cdot \gamma(u)]u = N(u) + \delta \cdot \Gamma(u). \quad (2.6)$$

Relationship (2.6) underscores the important role played by the integrated operators $L(u)$, $L^*(u)$, $\gamma(u)$, $\gamma^*(u)$: in contradistinction to the variational operators $N'(u)$, $N'^*(u)$, $\gamma'(u)$, $\gamma'^*(u)$, that are usually studied in relation to Eq. (2.1), it is the pair of integrated operators $\{L(u), \gamma(u)\}$ that restores exactly the original nonlinear system (2.1) when applied to u .

The backward (retarded) and forward (advanced) propagators are defined as the inverses of the operators $L(u) + \delta \cdot \gamma(u)$ and $L^*(u) + \delta^* \cdot \gamma^*(u)$, respectively:

$$[L(u) + \delta \cdot \gamma(u)]G_u = 1, \quad (2.7)$$

and

$$[L^*(u) + \delta^* \cdot \gamma^*(u)]G_u^* = 1, \quad (2.8)$$

where 1 denotes the unit operator. The Eqs. (2.7) and (2.8) can be written in terms of formal integral kernels as

$$[L(u(t)) + \delta \cdot \gamma(u(t))]G(u(t); t, t') = \delta(t - t'), \quad (2.9)$$

and

$$[L^*(u(t)) + \delta^* \cdot \gamma^*(u(t))]G^*(u(t); t, t'') = \delta(t - t''), \quad (2.10)$$

where t is a shorthand notation for the generic variable in the domain Ω (including its boundaries).

Since the operators $L(u) + \delta \cdot \gamma(u)$ and $L^*(u) + \delta^* \cdot \gamma^*(u)$ act linearly on the respective propagators, the relationships between the propagators and the expression for the solution u in terms of these propagators can be derived, as previously noted [3,4] in the same spirit as for the usual Green's functions in linear theory. Thus, forming the inner products of Eqs. (2.9) and (2.10) with $G^*(u(t); t, t'')$ and $G(u(t); t, t')$, respectively, and subtracting leads to the reciprocity relation

$$G^*(u(t); t, t') = G(u(t'); t', t). \quad (2.11)$$

The solution u of the original nonlinear system (2.1) is obtained in terms of the forward propagator G_u^* as follows:

$$\begin{aligned} u &= [\text{by Eq. (2.1)}] \\ &= \langle u, \delta \rangle - \langle N(u) + \delta \cdot \Gamma(u), G_u^* \rangle + \langle f + \delta \cdot g, G_u^* \rangle = [\text{by Eqs. (2.10) and} \\ & \quad (2.6)] \\ &= \langle u, [L^*(u) + \delta^* \cdot \gamma^*(u)]G_u^* \rangle - \langle [L(u) + \delta \cdot \gamma(u)]u, G_u^* \rangle + \langle f + \delta \cdot g, G_u^* \rangle \\ &= \langle f + \delta \cdot q, G_u^* \rangle, \end{aligned} \quad (2.12)$$

where the new surface term q contains the original surface term g and the surface terms arising from the respective integrations by parts of L^* and L over Ω . (Recall that L^* is the formal adjoint of L .) Using the reciprocity relationship (2.11) in Eq. (2.12) yields the solution u in terms of the backward propagator G_u as

$$u = \langle G_u, f + \delta \cdot q \rangle = \int_{\Omega} G(u(t); t, t') [f(t') + \delta \cdot q(t')] dt'. \quad (2.13)$$

3. APPLICATION OF THE NONLINEAR PROPAGATOR METHOD TO THE RICCATI EQUATION

We consider the Riccati equation

$$\frac{du}{dt} + bu^2 - c = 0, \quad (3.1)$$

subject to the initial condition

$$\lim_{t \rightarrow 0} u(t) = u_i, \quad (3.2)$$

where $t \in (0, t_f)$, and $b > 0$, $c > 0$, $u_i > 0$ are positive constants. A simple identification yields $N(u) = \frac{du}{dt} + bu^2$, $\delta \cdot \Gamma(u) = \delta(t) \cdot u(t)$, $f = c$, $g = u_i$. The unique solution of Eqs. (3.1) and (3.2) can readily be obtained by standard methods as

$$u(t) = \frac{u_i + (c/b)^{1/2} \tanh[t(bc)^{1/2}]}{1 + u_i(b/c)^{1/2} \tanh[t(bc)^{1/2}]} . \quad (3.3)$$

To apply the propagator method of Refs. [3–5] as sketched in Section 2, we consider that $u \in L_2([0, t_f])$ and calculate the Gâteaux-derivative, $N'(u)$, of $N(u)$ by applying the definition given in Eq. (2.2) to Eq. (3.1). This gives

$$N'(u)h = \frac{dh}{dt} + 2buh; \quad (3.4)$$

note that the operator $N'(u)$ acts linearly on h and depends parametrically on u .

The formal adjoint, $[N'(u)]^*$, of $N'(u)$ is readily obtained from Eq. (3.4) as

$$[N'(u)]^*v = -\frac{dv}{dt} + 2buv. \quad (3.5)$$

The operators $L(u)$ and $L^*(u)$ are obtained by applying Eqs. (2.4) and (2.5) to Eqs. (3.4) and (3.5), respectively. This yields

$$L(u)h = \frac{dh}{dt} + buh, \quad (3.6)$$

and

$$L^*(u)v = -\frac{dv}{dt} + buv. \quad (3.7)$$

As shown by Eqs. (2.12) and (2.13), the solution $u(t)$ of the Riccati equation can be obtained in terms of either the forward propagator $G_u^*(t, t')$ or the backward propagator $G_u(t, t')$. The forward propagator $G_u^*(t, t')$ is the solution of

$$L^*(u)G_u^*(t, t') = -\frac{dG_u^*(t, t')}{dt} + bu(t)G_u^*(t, t') = \delta(t - t'), \quad (3.8)$$

$$G_u^*(t, t') = 0, \quad \text{at } t = t_f \quad (\text{i.e., for } t > t'). \quad (3.9)$$

As expected, Eq. (3.8) is a (first-order, nonhomogeneous) linear equation for $G_u^*(t, t')$; its solution is

$$G_u^*(t, t') = H_+(t' - t) \exp \left[\int_{t'}^t bu(\tau) d\tau \right], \quad (3.10)$$

where $H_+(t' - t)$ is the unit step (Heaviside) function defined as

$$H_+(t' - t) = \begin{cases} 0, & \text{for } t' < t, \\ 1, & \text{for } t' \geq t. \end{cases} \quad (3.11)$$

The retarded propagator $G_u(t, t'')$ satisfies the linear system

$$L(u)G_u(t, t'') = \frac{dG_u(t, t'')}{dt} + bu(t)G_u(t, t'') = \delta(t - t''), \quad (3.12)$$

$$G_u(0, t'') = 0, \quad \text{at } t = 0 \quad (\text{i.e., for } t < t''). \quad (3.13)$$

The solution of Eqs. (3.12) and (3.13) is

$$G_u(t, t'') = H_+(t - t'') \exp \left[\int_t^{t''} bu(\tau) d\tau \right]. \quad (3.14)$$

As expected, the propagators G_u^* and G_u satisfy the reciprocity relationship (2.11).

The operations leading to Eq. (2.12) are performed as follows:

$$\begin{aligned} -u(t') &= \int_0^{t_f} \left(\frac{du}{dt} G_u^*(t, t') + bu^2 G_u^*(t, t') - cG_u^*(t, t') \right) dt \\ &\quad - \int_0^{t_f} \left(-u \frac{dG_u^*(t, t')}{dt} + u_2 b G_u^*(t, t') \right) dt \\ &= [u(t) G_u^*(t, t')]_0^{t_f} - \int_0^{t_f} c G_u^*(t, t') dt. \end{aligned}$$

This expression can be further reduced by using Eq. (3.9) and interchanging t and t' to obtain

$$u(t) = \int_0^{t_f} c G_u^*(t', t) dt' + u_i G_u^*(0, t). \quad (3.15)$$

Using the reciprocity relationship (2.12) in Eq. (3.15) gives

$$u(t) = \int_0^{t_f} c G_u(t, t') dt' + u_i G_u(t, 0). \quad (3.16)$$

Replacing the propagators G_u^* and G_u in Eqs. (3.15) and (3.16) by their respective expressions given by Eqs. (3.10) and (3.14) gives the following fixed-point form equation for $u(t)$:

$$u(t) = (\mathcal{N}u)(t),$$

where the nonlinear operator \mathcal{N} is given by

$$(\mathcal{N}u)(t) \equiv u_i e^{-\int_0^t bu(\tau)d\tau} + c \int_0^t dt' e^{-\int_{t'}^t bu(\tau)d\tau}. \quad (3.17)$$

By direct substitution, it is easy to verify that Eq. (3.16) satisfies the Riccati equation (3.1) and (3.2). The fixed-point Eq. (3.16) can be solved numerically by several methods. In the following, we solve it by the Picard iterative scheme, which is simply

$$u^{(n+1)}(t) = \mathcal{N}u^{(n)}(t), \quad (3.18)$$

where $u^{(0)}(t)$ is an initial guess, for example, $u_0(t) = u_i$.

To write a computer code for an arbitrary number of iterations, it is necessary to approximate the integrals in Eq. (3.17). For this purpose, we divide the time axis $[0, t_f]$, where t_f is the final time, into J equal closed intervals $[t_{j-1}, t_j]$, $j = 1, \dots, J$, where $t_j \equiv j\Delta$ and $\Delta \equiv t_f/J$. If Δ is chosen to be sufficiently small, the iterate $u^{(n)}(t)$ can be approximated by

$$u^{(n)}(t) \cong \left(\frac{u_j^{(n)} - u_{j-1}^{(n)}}{\Delta} \right) (t - t_{j-1}) + u_{j-1}^{(n)}, \quad t \in [t_{j-1}, t_j], \quad j = 1, \dots, J, \quad (3.19)$$

where $u_j^{(n)} \equiv u^{(n)}(t_j)$. Evaluating Eq. (3.18) at an arbitrary time step t_j gives

$$u_j^{(n+1)} = \mathcal{N}u^{(n)}(t_j). \quad (3.20)$$

The first integral in the definition of \mathcal{N} is discretized by replacing Eq. (3.19) in (3.18) and performing the respective integration over τ , i.e.,

$$b \int_0^{t_j} u^{(n)}(\tau)d\tau \cong b\Delta \sum_{i=1}^j \frac{u_i^{(n)} + u_{i-1}^{(n)}}{2}. \quad (3.21)$$

Using Eq. (3.19) in the second integral of $\mathcal{N}(t)$ in Eq. (3.17) leads to expressions involving the error function. Evaluating these expressions exactly would result in a substantial computational penalty; instead, we ignore the quadratic (in t) part of the exponential to obtain

$$\begin{aligned} \int_0^{t_j} dt' e^{-b \int_{t'}^{t_j} d\tau u^{(n)}(\tau)} &\cong \sum_{k=1}^j \left[\frac{e^{b\Delta u_{k-1}^{(n)}} - 1}{b u_{k-1}^{(n)}} \right] \\ &\times \exp \left[-b\Delta \sum_{i=k}^j \left(\frac{u_i^{(n)} + u_{i-1}^{(n)}}{2} \right) \right]. \end{aligned} \quad (3.22)$$

The approximation in (3.22) is justified a posteriori by the very high accuracy of the resulting solution. The expressions (3.22) and (3.21) are replaced in Eq. (3.20) to calculate iteratively the solution $u(t)$ to the Riccati equation. To avoid re-evaluating the sums and double sums at each time step, the equation that results by replacing (3.22) and (3.21) in Eq. (3.20) is

$$u_j^{(n+1)} = u_i \mathcal{S}_j^{(n)} + c \sum_{k=1}^j \mathcal{P}_{k,j}^{(n)}, \quad (3.23)$$

where

$$\begin{aligned} \mathcal{S}_j^{(n)} &= \mathcal{S}_{j-1}^{(n)} - \frac{b\Delta}{2} [u_j^{(n)} + u_{j-1}^{(n)}], \quad 0 < j \leq J, \\ \mathcal{S}_0^{(n)} &\equiv 0, \end{aligned} \quad (3.24)$$

$$\begin{aligned} \mathcal{P}_{k,j}^{(n)} &= \mathcal{P}_{k+1,j}^{(n)} + \mathcal{R}_{k,j}^{(n)} \left[\frac{e^{b\Delta u_{k-1}^{(n)}} - 1}{b u_{k-1}^{(n)}} \right], \quad 1 \leq k < j, \\ \mathcal{P}_{j,j}^{(n)} &= \mathcal{R}_{j,j}^{(n)} \left[\frac{e^{b\Delta u_{j-1}^{(n)}} - 1}{b u_{j-1}^{(n)}} \right], \end{aligned} \quad (3.25)$$

$$\begin{aligned} \mathcal{R}_{k,j}^{(n)} &= \mathcal{R}_{k+1,j}^{(n)} \times \exp \left[-\frac{b\Delta}{2} (u_k^{(n)} + u_{k-1}^{(n)}) \right], \quad 1 \leq k < j, \\ \mathcal{R}_{j,j}^{(n)} &= \exp \left[-\frac{b\Delta}{2} (u_j^{(n)} + u_{j-1}^{(n)}) \right]. \end{aligned} \quad (3.26)$$

4. APPLICATION OF THE DECOMPOSITION METHOD TO THE RICCATI EQUATION

As amply discussed in Ref. [1] and references therein, the decomposition method involves the separation of a nonlinear equation into a linear part that can be inverted, and expressing all remaining nonlinearities in a polynomial form. This separation is somewhat arbitrary and when the linear part is zero one has to create one by adding and subtracting a convenient linear operator. For the Riccati equation considered in Eqs. (3.1) and (3.2), one naturally identifies the linear and nonlinear operators with $\frac{d}{dt}$ and $c - bu^2$, respectively. The decomposition method leads to the following iterative expression for the solution $u(t)$:

$$u(t) = \sum_{n=0}^{\infty} u^{(n)}, \quad (4.1)$$

where

$$u^{(n+1)}(t) = \int_0^t f^{(n)}(s) ds, \quad (4.2)$$

and

$$f^{(n+1)}(t) = \frac{1}{(n+1)} \sum_{k=0}^n \left[(k+1) u^{(k+1)} \frac{\partial f^{(n)}}{\partial u^{(k)}} \right], \quad (4.3)$$

with starting values, for $n = 0$, given by

$$u^{(0)}(t) = u_i, \quad (4.4)$$

and

$$f^{(0)}(t) = c - bu_i^2. \quad (4.5)$$

The N th approximate to Eq. (4.1) is defined as

$$u_N(t) = \sum_{n=0}^N u^{(n)}(t). \quad (4.6)$$

To implement Eqs. (4.2) through (4.6) into a computer program for any N , it is necessary to derive a general expression for $u^{(n)}(t)$. Furthermore, the computational efficiency of such a code can be improved substantially if $u^{(n)}(t)$ could be written in a recursive form. It is easy to show that, for all $n > 0$, we can write $u^{(n)}(t)$ in the implicitly recursive form

$$u^{(n)} = \alpha^{(n)} t^n, \quad n > 0, \quad (4.7)$$

and

$$f^{(n)} = -b \sum_{k=0}^n u^{(k)} u^{(n-k)}, \quad n > 0. \quad (4.8)$$

where the coefficients $\alpha^{(n)}$ satisfy the recursive relation

$$\alpha^{(n+1)} \equiv -\frac{b}{(n+1)} \sum_{k=0}^n \alpha^{(k)} \alpha^{(n-k)}, \quad n > 0, \quad (4.9)$$

with

$$\alpha^{(0)} = u_i, \quad \alpha^{(1)} = (c - bu_i^2). \quad (4.10)$$

The recursion relationship (4.9) for the coefficients $\alpha^{(n)}$ is obtained by replacing Eqs. (4.7) and (4.8) in Eqs. (4.2) and (4.3), and performing the respective integration. The validity of Eqs. (4.7) and (4.8) can be readily verified by mathematical induction. Thus, the N th approximate, $u_N(t)$ to the solution $u(t)$ of the Riccati equation is obtained by substituting Eqs. (4.7) through (4.9) in Eq. (4.6); this gives

$$u_N(t) = u_i - b \sum_{n=1}^N \left(\frac{t^n}{n}\right) \sum_{k=0}^{n-1} \alpha^{(k)} \alpha^{(n-k-1)}. \quad (4.11)$$

5. NUMERICAL RESULTS AND CONCLUSIONS

This section presents the results of comparing the stability, accuracy, and computational efficiency of the nonlinear propagators methods coupled with the Picard iteration scheme, as given by Eq. (3.23), and the decomposition method, as given by Eq. (4.11). In the tables presenting the numerical results, these methods will be labeled as PROP-PIC and DECOMP, respectively. For comparison purposes, we will also include the results given by applying two standards methods—an implicit finite difference (FD) and a Picard iteration schemes—directly on the Riccati equation.

The finite difference (FD) scheme yields an expression for the solution at the current time step, u_j , in terms of the solution at the previous time step, u_{j-1} , as

$$u_j = \left[-1 \pm \sqrt{1 + 4b\Delta^2 (c + u_{j-1}/\Delta)} \right] / 2b\Delta, \quad j \geq 1. \quad (5.1)$$

The negative square root results in an unstable method; hence, only the positive sign in Eq. (5.1) was implemented in the computer program used to generate the numerical results. The evolutionary nature of Eq. (5.1), whereby the solution at any time step relates to the solution only at the immediately previous step, enabled us to use a time-stepping algorithm in our code.

To apply the Picard iteration to the Riccati equation, we first integrate Eq. (3.1) once to obtain

$$u(t) = \int_0^t f(u(\tau)) d\tau + u_i, \quad (5.2)$$

where

$$f(x) = c - bx^2. \quad (5.3)$$

Applying Picard's iterative method to Eq. (5.2) and evaluating the resulting expression at time step t_j gives

$$u_j^{(n)} = \int_0^{t_j} f(u^{(n-1)}(\tau)) d\tau + u_i, \quad j \geq 1, \quad n \geq 1. \quad (5.4)$$

This equation can be written in the recursive form

$$u_j^{(n)} = u_{j-1}^{(n)} + \int_{t_{j-1}}^{t_j} f(u^{(n-1)}(\tau)) d\tau. \quad (5.5)$$

The integral in Eq. (5.5) is evaluated using the trapezoidal formula on f itself so that

$$u_j^{(n)} = u_{j-1}^{(n)} + \frac{1}{2} [f(u_j^{(n-1)}) + f(u_{j-1}^{(n-1)})]. \quad (5.6)$$

In the following, the results given by Eq. (5.6) will be labeled INT-PIC (standing for "integral-Picard").

The comparative study of the stability, accuracy, and computational efficiency of the DECOMP, PROP-PIC, FD, and INT-PIC methods can be greatly simplified by rescaling Eqs. (3.1) and (3.2) so as to reduce the number of parameters from three (i.e., b, c , and u_i) to only one. This is achieved by introducing the scaling

$$\begin{aligned} v &\equiv \sqrt{b/c} u, \\ s &\equiv \sqrt{bc} t, \quad c \neq 0, \end{aligned} \quad (5.7)$$

to recast Eq. (3.1) into the equivalent form

$$dv/ds = 1 - v^2, \quad v(0) = v_i. \quad (5.8)$$

Thus, the numerical comparisons to follow will be done for Eq. (5.8) which is equivalent to Eq. (3.1) with $b = c = 1$; the initial conditions we have chosen are: $v_i = 1.1, 1.5, 2$, and 4 . The analytical solution of Eq. (5.8) is

$$V(s) = \frac{v_i + \tanh s}{1 + v_i \tanh s}. \quad (5.9)$$

The availability of the exact solution, Eq. (5.9), makes it possible to evaluate the relative pointwise accuracies of the numerical solutions obtained from the different methods. This accuracy is defined as

$$\mathcal{E} = \sup_{1 \leq j \leq J} \frac{|v_j - V(s_j)|}{V(s_j)}, \quad (5.10)$$

where $s_j \equiv \sqrt{bc} t_j$.

The final time step for each value of the initial condition was chosen so that the exact solution at s_J is within 10^{-3} of its steady state value $v = 1$ (i.e., $u = \sqrt{c/b}$). This time step, s_J , can be calculated from the approximate formula

$$s_J \cong \tanh^{-1} \left(1 - 10^{-3} \times \left| \frac{v_i + 1}{v_i - 1} \right| \right). \quad (5.11)$$

As will be seen in the tables below, not all methods were stable everywhere. However, for the unstable methods, which diverged numerically before reaching s_J , we experimented to obtain for each v_i the largest time level that the respective method can calculate before diverging. Also, the number of time steps, J , was chosen for each v_i so that a smaller J would result in an unacceptable accuracy; we set 10^{-4} to be the threshold for acceptable accuracy.

The results of our comparisons for the four initial conditions are shown in Tables 1 through 4. It is clear from these results that the DECOMP method is highly unstable. Even for the smallest initial condition, $v_i = 1.1$, the method diverges at $s = 1.5$, long before the specified final time $s = 2.3$. For the larger initial conditions, the method diverges at even smaller time levels. Instability of the DECOMP method likely stems from the secular terms that it generates. For final time levels where the DECOMP method is stable, the results produced are

quite accurate, even though the method takes a very large number of iterations to converge.

In contradistinction to the DECOMP method, the PROP-PIC method is stable for all initial conditions and final time levels considered here. This method achieved the specified level of accuracy with rather large time steps (especially when compared to the implicit FD method) and in a relatively small number of iterations. The stability of this method has been predicted earlier [1,2] because the secular terms appear only in their summed form, which is a decaying exponential.

The implicit FD method is stable, as expected on theoretical grounds, for all cases considered; however, the larger initial conditions require extremely large numbers of time steps in order to achieve the specified accuracy. This, of course, reflects very heavily on the CPU time requirements (e.g., between ten and twenty times that required by the PROP-PIC method). The INT-PIC method is unstable, but less so than the DECOMP method—in the sense that for the same initial condition, the INT-PIC method can reach a larger time level before diverging. This method is inferior to the PROP-PIC and the implicit FD methods because it is unstable at large time levels for $v_i \geq 1.5$. When it converges, the INT-PIC method requires a moderate number of time steps to achieve the specified accuracy. For the case $v_i = 1.1$, when the INT-PIC method converged over the entire specified time interval, it required three times as many time steps, almost twice as many iterations, and slightly more CPU time than the PROP-PIC method to achieve comparable accuracy. Even for the case $v_i = 1.5$, the PROP-PIC method required fewer time steps, fewer iterations, and only a slightly larger CPU time, even though the INT-PIC method diverged beyond $s = 2.7$.

Based on the foregoing discussion of numerical results, we conclude that the method of nonlinear propagators is at least as efficient and accurate as the finite difference or straightforward Picard iteration method, while the decomposition method diverges quickly—being useful only for small time values. Of course, the Riccati equation—a simple ordinary differential equation—considered in this work is not a definitive benchmark for judging the comparative efficiency and accuracy of either the method of decomposition or the method of nonlinear propagators. These methods should be benchmarked on initial/boundary value problems involving nonlinear partial differential equations. A first step in this direction for benchmarking the method of nonlinear propagators has recently been taken by Cacuci and Karakashian [2], who report that for the Korteweg-de Vries equation this method is superior to all but one (a finite element method, which is of *equivalent* efficiency) of the methods presently available for solving this equation.

METHOD	ACCURACY	CPU (sec)	NUMBER ITERATIONS	NUMBER TIME STEPS	COMMENTS
DECOMP	.3E-4	.181	624	10 (final time = 1.5)	Unstable for final time ≥ 1.6 Number time steps = 10,000
PROP-PIC	.4E-4	.128	9	10	Stable
F. D.	.9E-4	.972	NA	900	Stable
INT-PIC	.8E-4	.144	16	30	Stable

Table 1. Comparison of Results of the Four Methods for
Initial Condition $v_i = 1.1$ and Final Time $s = 2.3$ Unless Stated Otherwise

METHOD	ACCURACY	CPU (sec)	NUMBER ITERATIONS	NUMBER TIME STEPS	COMMENTS
DECOMP	.8E-4	.625	1514	10 (final time = .8)	Unstable for final time $\geq .9$ Number time steps = 10,000
PROP-PIC	.8E-4	.190	12	45	Stable
F. D.	.9E-4	6.458	NA	6500	Stable
INT-PIC	.9E-4	.165	20	100 (final time = 2.7)	Unstable for final time ≥ 2.8 Number of time steps = 10,000

Table 2. Comparison of Results of the Four Methods for
Initial Condition $v_i = 1.5$ and Final Time $s = 3.0$ Unless Stated Otherwise

METHOD	ACCURACY	CPU (sec)	NUMBER ITERATIONS	NUMBER TIME STEPS	COMMENTS
DECOMP	.2E-4	.179	606	10 (final time = .54)	Unstable for final time $\geq .6$ Number time steps = 10,000
PROP-PIC	.8E-4	.412	13	95	Stable
F. D.	.9E-4	12.653	NA	15000	Stable
INT-PIC	.9E-4	.184	20	130 (final time = 2.3)	Unstable for final time ≥ 2.4 Number of time steps = 10,000

Table 3. Comparison of Results of the Four Methods for Initial Condition $v_i = 2.0$ and Final Time $s = 3.3$ Unless Stated Otherwise

METHOD	ACCURACY	CPU (sec)	NUMBER ITERATIONS	NUMBER TIME STEPS	COMMENTS
DECOMP	.3E-4	.099	163	10 (final time = .24)	Unstable for final time $\geq .3$ Number time steps = 10,000
PROP-PIC	.9E-4	2.661	15	270	Stable
F. D.	.9E-4	36.317	NA	50000	Stable
INT-PIC	.4E-4	.178	15	210 (final time = .9)	Unstable for final time ≥ 1.0 Number of time steps = 10,000

Table 4. Comparison of Results of the Four Methods for
Initial Condition $v_i = 4.0$ and Final Time $s = 3.6$ Unless Stated Otherwise

REFERENCES

1. G. Adomian, "Nonlinear Stochastic Operator Equations," Academic Press, Inc., Orlando, Florida, 1986.
2. D. G. Cacuci and O. A. Karakashian, "A New Method for Solving the Generalized Burgers-Korteweg-de Vries Equation," *J. Comp. Phys.*, submitted.
3. D. G. Cacuci, R. B. Perez, and V. Protopopescu, *Proc. IVth International Workshop on Nonlinear Evolution Equations and Dynamical Systems*, Balaruc-Les-Bains, June 11-26, 1987.
4. D. G. Cacuci, R. B. Perez, and V. Protopopescu, "Duals and Propagators: A Canonical Formalism for Nonlinear Equations," *J. Math. Phys.* **29**, No. 2, (1988) 353-361.
5. D. G. Cacuci and V. Protopopescu, "Propagators for Nonlinear Systems," *J. Phys. A*, submitted.

INTERNAL DISTRIBUTION

- | | |
|---|------------------------------------|
| 1. B. R. Appleton | 18. Laboratory Records,
ORNL-RC |
| 2-6. Y. Y. Azmy | 19. Document Reference
Section |
| 7. D. T. Ingersoll | 20. Central Research Library |
| 8. F. C. Maienschein | 21. ORNL Patent Section |
| 9-13. V. Protopopescu | |
| 14. R. T. Santoro | |
| 15. EPMD Reports Office | |
| 16-17. Laboratory Records
Department | |

EXTERNAL DISTRIBUTION

- 22-26. Dan G. Cacuci, Chemical and Nuclear Engineering Department, 3355 Engineering II, University of California at Santa Barbara, Santa Barbara, California 93106
27. J. J. Dorning, Department of Nuclear Engineering and Engineering Physics, Thornton Hall, University of Virginia, Charlottesville, Virginia 22901
28. R. M. Haralick, Department of Electrical Engineering, University of Washington, Seattle, Washington 98195
29. James E. Leiss, 13013 Chestnut Oak Drive, Gaithersburg, Maryland 20878
30. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, Illinois 61801
31. Mary F. Wheeler, Mathematics Department, University of Houston, 4800 Calhoun, Houston, Texas 77204-3476
32. Office of the Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations, P.O. Box 2001, Oak Ridge, TN 37831
- 33-42. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, Tennessee 37830