Adomian Decomposition Method:
Convergence Analysis and Numerical Approximations

By

Ahmed H M Abdelrazec

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2008)	McMaster University
(Mathematics)	Hamilton, Ontario

TITLE:	Adomian Decomposition Method: Convergence Analysis and Numerical Approximations
AUTHOR:	Ahmed H M Abdelrazec
SUPERVISOR:	Professor. Dmitry Pelinovsky
NUMBER OF PAGES:	v, 53

Abstract

We prove convergence of the Adomian Decomposition Method (ADM) by using the Cauchy-Kovalevskaya theorem for differential equations with analytic vector fields, and obtain a new result on the convergence rate of the ADM. Picard's iterative method is considered for the same class of equations in comparison with the decomposition method. We outline some substantial differences between the two methods and show that the decomposition method converges faster than the Picard method. Several non-linear differential equations are considered for illustrative purposes and the numerical approximations of their solutions are obtained using MATLAB. The numerical results show how the decomposition method is more effective than the standard ODE solvers. Moreover, we prove convergence of the ADM for the partial differential equations and apply it to the cubic nonlinear Schrödinger equation with a localized potential.

Acknowledgments

I would like to express my deepest gratitude to supervisor, Professor Dmitry Pelinovsky, without whose support and guidance, the completion of this thesis would have been impossible. Professor Pelinovsky has spent a great deal of his time with my drafts and helped me hammer them into a coherent shape. From the day I became a graduate student in McMaster University, he has inspired and taught me the powerful tools of applied mathematics. I thank him for his critical insights, enormous patience, sense of perspective and his continuous encouragement. Grateful acknowledgment is also made to Professor Nicholas Kevlahan for his kindness, generosity and support. Productive discussions at the AIMS LAB Seminar at McMaster University have greatly illuminated the ideas for my work and have given me a backbone in analysis. So I would like to thank the members of the AIMS LAB for their contributions, in particular Professors Stanley Alama and Bartosz Protas for careful reading of my thesis and useful corrections. I also extend my appreciation to all the staff members at the Mathematics Department for their competent guidance and advice through the administrative corridors of the university.

I owe particular debt to my dear parents and family. My gratitude to my wife, Sherine, and our daughter, Noura, remains more than anything I can put into words as they have always enriched my life with the foundation of tranquility and love.

Contents

Introduction

In the 1980's, George Adomian (1923-1996) introduced a new powerful method for solving nonlinear functional equations. Since then, this method has been known as the Adomian decomposition method (ADM) [3,4]. The technique is based on a decomposition of a solution of a nonlinear operator equation in a series of functions. Each term of the series is obtained from a polynomial generated from an expansion of an analytic function into a power series. The Adomian technique is very simple in an abstract formulation but the difficulty arises in calculating the polynomials and in proving the convergence of the series of functions.

Convergence of the Adomian method when applied to some classes of ordinary and partial differential equations is discussed by many authors. For example, K. Abbaoui and Y. Cherruault [1,2] proved the convergence of the Adomian method for differential and operator equations. Lesnic [35] investigated convergence of the ADM when applied to time-dependent heat, wave and beam equations for both forward and backward time evolution. He showed that the convergence was faster for forward problems than for backward problems. Al-Khaled and Allan [7] implemented the Adomian method for variable-depth shallow water equations with a source term and illustrated the convergence numerically. A comparative study between the ADM and the Sinc-Galerkian method for solving population growth models was performed by Al-Khaled [6], while that between the ADM and the Runge Kutta method for solving systems of ordinary differential equations was performed by Shawagfeh and Kaya [42]. Wazwaz and Khuri discussed applications of the ADM to a class of Fredholm integral equations that occurs in acoustics [44]. Wazwaz also compared the ADM and the Taylor series method by using some particular examples, and showed that the decomposition method produced reliable results with fewer iterations, whereas the Taylor series method suffered from computational difficulties [45]. In [46] Wazwaz modified the ADM to accelerate the convergence of the series solution. The validity of the modified technique was verified through illustrative examples. Furthermore, in [47] he developed a numerical algorithm to approximate solutions of higher-order boundary-value problems. Application of Chebyshev polynomials to numerical implementation of the ADM were discussed by Hosseini [29].

In [25] Guellal and Cherruault used the Adomian's technique for solving an elliptic boundary value problem with an auxiliary condition. Ndour et al. [38] used the decomposition method to solve the system of differential equations governing the interaction model of two species. Comparative study between the Adomian method and wavelet-Galerkin method for solving integro-differential equations was performed by El-Sayed and Abdel-Aziz [19]. El-Sayed and Gaber used the Adomian method for solving partial differential equation of fractal order in a finite domains [18]. Adomian et al. [5] used the technique to solve mathematical models of the immune response to a population of bacteria, viruses, antigens or tumor cells that are expressed by systems of nonlinear differential equations or delay-differential equations. Laffez and Abbaoui [34] studied a model of thermic exchanges in a drilling well with the decomposition method. Guellal et al. [26] used the decomposition method for solving differential systems coming from physics and compared it to the Runge-Kutta method. Sanchez et al. [41] investigated the weaknesses of the thin-sheet approximation and proposed a higher-order development allowing to increase the range of convergence and preserve the nonlinear dependence of the variables. Edwards et al. [17] compared the ADM and the Runge-Kutta methods for approximate solutions of predator prey model equations.

Jafari and Gejji [32] modified the ADM to solve a system of nonlinear equations. They obtained a series solution with a faster convergence than the one obtained by the standard ADM. Luo et al. [36] revised the ADM for cases involving inhomogeneous boundary conditions, using a suitable transformation. Luo [37] proposed an efficient modification to the ADM, namely a two-step Adomian Decomposition Method that facilitated the calculations. Zhang [49] presented a modified ADM to solve a class of nonlinear singular boundary-value problems, which arise as normal model equations in nonlinear conservative systems. Zhu et al. [50] presented a new algorithm for calculating Adomian polynomials for nonlinear operators. Gejji and Jafari [21] presented an iterative method for solving nonlinear functional equations. In addition, the ADM was used to solve a wide range of physical problems in various engineering fields such as vibration and wave equation [9] and [15], porous media simulation [39], fluid flow [8], heat and mass transfer [16].

Thus, we see that the Adomian decomposition method has been used to solve many functional and differential equations so far. The purpose of this thesis is to study convergence and stability of this method in application to the initial-value problems for systems of nonlinear differential equations. We prove convergence of the ADM by using the Cauchy-Kovalevskaya theorem for differential equations with analytic vector fields, and obtain a new result on the convergence rate of the ADM. Picard's iterative method is considered for the same class of equations in comparison with the decomposition method. We outline some substantial differences between the two methods and show that the decomposition method converges faster than the Picard method. Several nonlinear differential equations are considered for illustrative purposes and the numerical approximations of their solutions are obtained using MATLAB. The numerical results show how the decomposition method is more effective than the standard ODE solvers. Moreover, we prove convergence of the ADM for the partial differential equations and apply it to the cubic nonlinear Schrödinger equation with locaized potential.

This thesis is structured as follows: Chapter 1 is devoted to convergence of the ADM for ordinary differential equations. It consists of five sections. The Adomian decomposition method is described in Section 1.1. A comparison between the ADM and the Picard method is demonstrated in Section 1.2. Section 1.3 gives a simple proof of convergence of the Adomian technique by using the Cauchy-Kovalevskaya theorem. The rate of convergence of the ADM is studied in Section 1.4. Section 1.5 presents a counter example to prove that the ADM is not a contraction method.

Chapter 2 is devoted to numerical implementation of the ADM in MATLAB. It consists of two sections. Section 2.1 formulates numerical algorithms for the ADM and the Picard method in application to initial-value problems for ODE's. Two examples of second-order differential equations are presented in Section 2.2 to illustrate the accuracy of the ADM.

Chapter 3 extends the convergence analysis and numerical approximations to partial differential equations. It consists of two sections. Section 3.1 gives a proof of convergence of the ADM for semilinear PDEs associated to an unbounded differential operator. Section 3.2 presents two numerical examples of solutions of the cubic nonlinear Schrödinger equation with a localized potential.

Chapter 1

Convergence of the ADM for ODEs

In this chapter, we prove convergence of the ADM for initial-value problems associated with systems of ordinary differential equations.

1.1 Formalism of the ADM

In reviewing the basic methodology, we consider an abstract system of nonlinear differential equations:

$$\frac{dy}{dt} = f(t, y), \quad y \in \mathbb{R}^d, \quad f : \mathbb{R} \times \mathbb{R}^d \longrightarrow \mathbb{R}^d, \tag{1.1}$$

with initial condition $y(0) = y_0 \in \mathbb{R}^d$. Assume that f is analytic near $y = y_0$ and $t = 0$. It is equivalent to solve the initial value problem for (1.1) and the Volterra integral equation

$$y(t) = y_0 + \int_0^t f(s, y(s))ds. \tag{1.2}$$

To set up the Adomian method, consider y in the series form:

$$y = y_0 + \sum_{n=1}^{\infty} y_n, \tag{1.3}$$

and write the nonlinear function $f(t,y)$ as the series of functions,

$$f(t,y) = \sum_{n=0}^{\infty} A_n \left(t, y_0, y_1, ..., y_n\right). \tag{1.4}$$

The dependence of A_n on t and y_0 may be non-polynomial. Formally, A_n is obtained by

$$A_n = \frac{1}{n!}\frac{d^n}{d\varepsilon^n} f\left(t, \sum_{i=0}^{\infty} \varepsilon^i y_i\right)\Bigg|_{\varepsilon=0}, \quad n = 0, 1, 2, ... \tag{1.5}$$

where ε is a formal parameter. Functions A_n are polynomials in $(y_1, ..., y_n)$, which are referred to as the Adomian polynomials.

In what follows, we shall consider a scalar differential equation and set $d = 1$. A generalization for $d \geqslant 2$ is possible but is technically longer.

The first four Adomian polynomials for $d = 1$ are listed as follows:

$$\begin{aligned}
A_0 &= f(t, y_0) \\
A_1 &= y_1 f'(t, y_0) \\
A_2 &= y_2 f'(t, y_0) + \frac{1}{2} y_1^2 f''(t, y_0) \\
A_3 &= y_3 f'(t, y_0) + y_1 y_2 f''(t, y_0) + \frac{1}{6} y_1^3 f'''(t, y_0),
\end{aligned}$$

where primes denote the partial derivatives with respect to y.

It was proven by Abbaoui and Cherruault [1,2] that the Adomian polynomials A_n are defined by the explicit formulae:

$$A_n = \sum_{k=1}^{n} \frac{1}{k!} f^{(k)}(t, y_0) \left(\sum_{p_1+...+p_k=n} y_{p_1} .... y_{p_k} \right), \quad n \geq 1,$$

or, in an equivalent form, by

$$A_n = \sum_{|nk|=n} f^{(|k|)}(t, y_0) \frac{y_1^{k_1} .... y_n^{k_n}}{k_1! .... k_n!}, \quad n \geq 1,$$

where $|k| = k_1 + ... + k_n$, and $|nk| = k_1 + 2k_2 + ... + nk_n$.

Khelifa and Cherruault [33] proved a bound for Adomian polynomials by,

$$|A_n| \leqslant \frac{(n+1)^n}{(n+1)!} M^{n+1}, \tag{1.5}$$

where

$$\sup_{t \in J} \left| f^{(k)}(t, y_0) \right| \leqslant M,$$

for a given time interval $J \subset \mathbb{R}$.

Substitution of (1.3) and (1.4) into (1.2) gives a recursive equation for y_{n+1} in terms of $(y_0, y_1, ..., y_n)$:

$$y_{n+1}(t) = \int_0^t A_n(s, y_0(s), y_1(s), ..., y_n(s)) ds, \quad n = 0, 1, 2, ... \tag{1.6}$$

Convergence of series (1.3) obtained by (1.6) is a subject of our studies in this chapter.

1.2 Comparison between the ADM and the Picard method

The ADM was first compared with the Picard method by Rach [40] and Bellomo and Sarafyan [12] on a number of examples. Golberg [22] showed that the Adomian method for linear differential equations was equivalent to the classical method of successive approximations (Picard iterations). However, this equivalence does not hold for nonlinear differential equations. In this section we compare the two methods and show differences and advantages of the decomposition method.

Recall that Picard's method introduced by Emile Picard in 1891, is used for the proof of existence and uniqueness of solutions of a system of differential equations. The Picard method starts with analysis of Volterra's integral equation (1.2). Assume that $f(t, y)$ satisfies a local Lipschitz condition in a ball around $t = 0$ and $y = y_0$:

$$\forall |t| \leq t_0, \quad \forall |y - y_0|, |\tilde{y} - y_0| \leq \delta_0 : \quad |f(t, y) - f(t, \tilde{y})| \leq K |y - \tilde{y}|,$$

where K is Lipshitz constant and $|y|$ is any norm in $\mathbb{R}^d$, e.g. the Euclidean norm $|y| = (y_1^2 + ... + y_d^2)^{\frac{1}{2}}$.

Let $y^{(0)} = y_0$ and define a recurrence relation

$$y^{(n+1)}(t) = y_0 + \int_0^t f(s, y^{(n)}(s))ds, \quad n = 0, 1, 2, ... \tag{1.7}$$

If t_0 is small enough, the new approximation $y^{(n+1)}(t)$ belongs to the same ball $|y - y_0| \leq \delta_0$ for all $|t| \leq t_0$ and the map (1.7) is a contraction in the sense that

$$\left| \int_0^t [f(s, y(s)) - f(s, \tilde{y}(s))] \, ds \right| \leq Q \sup_{|t| \leq t_0} |y(t) - \tilde{y}(t)|, \tag{1.8}$$

where $Q = Kt_0 < 1$, so that $t_0 < \dfrac{1}{K}$. By the Banach fixed point theorem, there exists a unique solution $y(t)$ in $C([-t_0, t_0], B_{\delta_0}(y_0))$ where $B_{\delta_0}(y_0)$ is an open ball in $\mathbb{R}^d$ centered at y_0 with radius δ_0. Recall here that $C([-t_0, t_0], \mathbb{R}^d)$ with the norm

$$\|y\| = \sup_{|t| \leq t_0} |y(t)| \tag{1.9}$$

is a complete metric space. Since the integral of a continuous function is a continuously differintiable function, $y(t)$ is actually in $C^1([-t_0, t_0], B_{\delta_0}(y))$. By the contraction mapping principle, the error of the approximate solution $y^{(n)}(t)$ is estimated by:

$$E_n = \left\| y - y^{(n)} \right\| \leqslant \frac{MK^n t_0^{n+1}}{(n+1)!}, \quad M = \sup_{|t| \leq t_0} \sup_{|y - y_0| \leq \delta_0} |f(t, y)|.$$

In [30], Hosseini and Nasabzadeh claimed that the Adomian iteration method (1.2) can be formulated as

$$Y_{n+1} = y_0 + \int_0^t f(s, Y_n(s))ds, \tag{1.10}$$

where

$$Y_n = y_0 + \sum_{k=1}^{n} y_k. \tag{1.11}$$

However, the claim is wrong since

$$\sum_{i=0}^{n} A_i(t, y_0, y_1, ..., y_i) \neq f(t, Y_n(t)), \quad n \geq 1.$$

Moreover, the above iteration formula on $Y_n, n \in \mathbb{N}$ is nothing but Picard's iteration formula and, therefore, the proof of convergence of the iterative method (1.10) in [30] repeates the standard proof of convergence of Picard iterations and gives no proof of convergence of the ADM. Computations of Picard's iterative algorithm were reported recently in [48].

Now, we shall understand the relationship between the ADM and the Picard method using an example of a scalar first order ODE:

$$\begin{cases} \dfrac{dy}{dt} = y^p \\ y(0) = 1 \end{cases} \tag{1.12}$$

where $p \geqslant 1$. This differential equation has the exact solution

$$y(t) = \frac{1}{(1-(p-1)t)^{\frac{1}{p-1}}} \tag{1.13}$$

Following the Adomian method, we write

$$y(t) = 1 + \int_0^t y^p(s)ds$$

and compute the Adomian polynomials from $f = y^p$ in the form:

$$\begin{aligned} A_0 &= y_0^p, \\ A_1 &= p y_0^{p-1} y_1, \\ A_2 &= \frac{p(p-1)}{2} y_0^{p-2} y_1^2 + p y_0^{p-1} y_2 \\ A_3 &= \frac{p(p-1)(p-2)}{6} y_0^{p-3} y_1^3 + p(p-1) y_0^{p-2} y_1 y_2 + p y_0^{p-1} y_3 \end{aligned}$$

Using (1.6), we determine few terms of the Adomian series:

$$\begin{aligned} y_0(t) &= 1, \\ y_1(t) &= t, \\ y_2(t) &= \frac{p}{2} t^2, \\ y_3(t) &= \frac{p(2p-1)}{3!} t^3, \\ y_4(t) &= \frac{p(6p^2-7p+2)}{4!} t^4. \end{aligned}$$

Expanding (1.13) in a power series of t, we can see that the Adomian decomposion recovers the power series solution:

$$\begin{aligned} y(t) &= \frac{1}{(1-(p-1)t)^{\frac{1}{p-1}}} \\ &= 1 + t + \frac{p}{2}t^2 + \frac{p(2p-1)}{3!}t^3 + \frac{p(6p^2-7p+2)}{4!}t^4 + O(t^5) \\ &= y_0 + y_1 + y_2 + y_3 + y_4 + \ldots \end{aligned}$$

On the other hand, using Picard iterations,

$$y^{(n+1)} = 1 + \int_0^t \left(y^{(n)}(s)\right)^p ds,$$

we obtain succesive approximations in the form:

$$\begin{aligned} y^{(0)} &= 1, \\ y^{(1)} &= 1 + t, \\ y^{(2)} &= \frac{p}{1+p} + \frac{(1+t)^{1+p}}{1+p}, \\ y^{(3)} &= 1 + \int_0^t \left(y^{(2)}\right)^p ds. \end{aligned}$$

Starting with $y^{(2)}$, Picard approximations mix up powers of t which make $y^{(n)}$ being different from the n-th partial sum of the power series. For instance, if $p = 2$, then

$$\begin{aligned} y^{(0)} &= 1 = y_0, \\ y^{(1)} &= 1 + t = y_0 + y_1, \\ y^{(2)} &= 1 + t + t^2 + \frac{t^3}{3} = y_0 + y_1 + y_2 + \frac{t^3}{3} \\ y^{(3)} &= 1 + t + t^2 + t^3 + \frac{2}{3}t^4 + \frac{1}{3}t^5 + \frac{t^6}{9} + \frac{t^7}{63} \\ &= y_0 + y_1 + y_2 + y_3 + \frac{2}{3}t^4 + \frac{1}{3}t^5 + \frac{t^6}{9} + \frac{t^7}{63}. \end{aligned}$$

Since $\sum_{i=0}^{n} y_i(t)$ is a partial sum of the power series for (1.13), we conclude that the Adomian method better approximates the exact power series solution compared to the Picard method. In general, since the Adomian method requires analyticity of $f(t, y)$, which is more restictive than the Lipschitz condition required for the Picard method, we expect that the ADM converges faster than the Picard method. We will illustrate this feature in Chapter 2.

1.3 Convergence Analysis

It is clear from (1.5) that A_n are polynomials in $y_1, ..., y_n$ and thus y_{n+1} is obtained from (1.6) explicitly, if we are able to calculate A_n. The first proof of convergence of the ADM was given by Cherruault [14], who used fixed point theorems for abstract functional equations. Furthermore, Babolian and Biazar [10] introduced the order of convergence of the ADM, and Boumenir and Gordon [13] discussed the rate of convergence of the ADM.

The proof of the convergence for the ADM was discussed by Cherruault [14] (see also Himoun, Abbaoui, and Cherruault [27,28] for recent results in the context of the functional equation

$$y = y_0 + f(y), \quad y \in \mathbb{H}, \tag{1.14}$$

where $\mathbb{H}$ is a Hilbert space and $f : \mathbb{H} \to \mathbb{H}$. Let $S_n = y_1 + y_2 + ... + y_n$, and $f_n(y_0 + S_n) = \sum_{i=0}^{n} A_i$. The ADM is equivalent to determining the sequence $\{S_n\}_{n\in\mathbb{N}}$ defined by

$$S_{n+1} = f_n(y_0 + S_n), \quad S_0 = 0$$

If there exist limits

$$S = \lim_{n\to\infty} S_n, \quad f = \lim_{n\to\infty} f_n$$

in a Hilbert space $\mathbb{H}$, then S solves a fixed-point equation $S = f(y_0 + S)$ in $\mathbb{H}$. The convergence of the ADM was proved in [14], under the following two conditions:

$$\|f\| \leqslant 1, \quad \|f_n - f\| = \varepsilon_n \longrightarrow 0 \quad as \quad n \longrightarrow \infty \tag{1.15}$$

These two conditions are rather restrictive. The first condition implies a constraint on the nonlinear function (1.14) while, the second condition implies the convergence of the series $\sum_{n=0}^{\infty} A_n$. It is difficult to satisfy the two conditions for a given nonlinear function $f(y)$. In the following, we shall prove convergence of the Adomian method in the context of the ODE systems (1.1) by using the Cauchy-Kovalevskaya theorem. We only require that the nonlinearity f be analytic in t and y. Let us start by reviewing the Cauchy-Kovalevskaya theorem for ordinary differential equations.

Theorem 1.3.1. *Let $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ be a real analytic function in the domain $[-t_0, t_0] \times B_{\delta_0}(y_0)$ for some $t_0 > 0$ and $\delta_0 > 0$. Let $y(t; y_0)$ be a unique solution for $t \in [-t_0, t_0]$ of the initial-value problem*

$$\begin{cases} \dfrac{dy}{dt} = f(t, y) \\ y(0) = y_0 \end{cases} \tag{1.16}$$

Then $y(t; y_0)$ is also a real analytic function of t near $t = 0$ that is there exists $\tau \in (0, t_0)$ such that $y : [-\tau, \tau] \to \mathbb{R}^d$ is a real analytic function.

Remark 1.3.2. *Existence, uniqueness and continuous dependence on t and y_0 of $y(t; y_0)$ follows from Picard's method since if f is real analytic, then it is locally Lipschitz.*

Remark 1.3.3. *We shall consider and prove Theorem 1.3.1 for $d = 1$. Generalization for $d \geq 2$ can be developed with a more complicated formalism, see [43] for further details on Cauchy-Kovalevskaya theorem.*

Proof. By Cauchy estimates for a real analytic function in the domain $[t_0, t_0] \times B_{\delta_0}(y_0)$ [24], there exist $a, C > 0$ such that

$$1 + |f(0, y_0)| \leq C \tag{1.17}$$

$$\sum_{k_1+k_2=k} \frac{1}{k_1! k_2!} \left|\partial_t^{k_1} \partial_y^{k_2} f(0, y_0)\right| \leq \frac{C}{a^k}, \quad \forall k \geq 1, k_1, k_2 \geq 0 \tag{1.18}$$

By the Cauchy estimates (1.17, 1.18), the Taylor series for $f(t,y)$ at $t=0$ and $y=y_0$ is bounded by

$$1+|f(t,y)| \leq C\sum_{k=0}^{\infty}\left(\frac{\rho}{a}\right)^k = \frac{C}{1-\frac{\rho}{a}} = \frac{Ca}{a-\rho} = g(\rho),$$

where $\rho = |t|+|y-y_0| < a$. By the Weierstrass M-test, the Taylor series for f converges for all

$$|t|+|y-y_0| < a. \tag{1.19}$$

Therefore, we have

$$\begin{aligned}
1+|f(0,y_0)| &\leq C = g(0)\\
\sum_{k_1+k_2=k}\frac{1}{k_1!k_2!}\left|\partial_t^{k_1}\partial_y^{k_2}f(0,y_0)\right| &\leq \frac{C}{a^k} \equiv \frac{1}{k!}g^{(k)}(0), \quad \forall k_1,k_2\geq 0, k\geq 1.
\end{aligned}$$

Let us consider a majorant problem for $\rho \in \mathbb{R}_+$:

$$\begin{cases} \dfrac{d\rho}{dt} = g(\rho) = \frac{Ca}{a-\rho} \\ \rho(0) = 0 \end{cases}$$

This problem has an explicit solution

$$\rho(t) = a - \sqrt{a^2 - 2aCt},$$

which is an analytic function of t in $|t| < \frac{a}{2C}$. By comparison principle, if

$$\begin{cases} \dfrac{dy}{dt} = f(t,y) \\ y(0) = y_0 \end{cases}$$

and $1+|f(t,y)| \leq g(|t|+|y(t)-y_0|)$, for all $|t|+|y(t)-y_0| < a$ then

$$|t|+|y(t)-y_0| \leq \rho(t) = a-\sqrt{a^2-2aCt} = \sum_{k=1}^{\infty}\frac{1}{k!}\rho^{(k)}(0)t^k.$$

Therefore, for all $t\geq 0$,

$$|y(t;0)-y_0| \leq t(\rho'(0)-1) + \sum_{k\geq 2}\frac{1}{k!}\rho^{(k)}(0)t^k,$$

where the Taylor series converges absolutely in $|t| < \frac{a}{2C}$. To prove that $y(t; y_0)$ is analytic function in $|t| < \min\left(a, \frac{a}{2C}\right)$, it remains to prove that $\left|y^{(k)}(0, y_0)\right| \leq \rho^{(k)}(0)$ for any $k \geq 1$. If this is the case, then the Taylor series for $y(t, y_0)$ has a majorant convergent series, such that the Taylor series for $y(t, y_0)$ converges, by the Weierstrass M-Test. To prove that $\left|y^{(k)}(0; y_0)\right| \leq \rho^{(k)}(0)$, we compute the first three derivatives explicitly from the ODE system:

$$
\begin{aligned}
\frac{d^2y}{dt^2} &= \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} = \frac{\partial f}{\partial t} + f\frac{\partial f}{\partial y} \\
\frac{d^3y}{dt^3} &= \frac{\partial^2 f}{\partial t^2} + 2f\frac{\partial^2 f}{\partial t \partial y} + \frac{\partial f}{\partial t}\frac{\partial f}{\partial y} + f\left(\frac{\partial f}{\partial y}\right)^2 + f^2\frac{\partial^2 f}{\partial y^2}
\end{aligned}
$$

for example

$$
\begin{aligned}
\left|\frac{d^2y}{dt^2}\right| &\leq \left|\frac{\partial f}{\partial t}\right| + \left|\frac{\partial f}{\partial y}\right||f| \\
&\leq g'(0)\left(1 + |f|\right) \leq g(0)g'(0) = \frac{d^2\rho}{dt^2}(0)
\end{aligned}
$$

$$
\begin{aligned}
\left|\frac{d^3y}{dt^3}\right| &\leq \left|\frac{\partial^2 f}{\partial t^2}\right| + 2\left|\frac{\partial^2 f}{\partial y \partial t}\right||f| + \left|\frac{\partial f}{\partial t}\right|\left|\frac{\partial f}{\partial y}\right| + |f|\left|\frac{\partial f}{\partial y}\right|^2 + |f|^2\left|\frac{\partial^2 f}{\partial y^2}\right| \\
&\leq g''(0)\left(1 + |f|\right)^2 + \left(g'(0)\right)^2\left(1 + |f|\right) \\
&\leq g^2(0)g''(0) + \left(g'(0)\right)^2 g(0) = \frac{d^3\rho}{dt^3}(0)
\end{aligned}
$$

Generally

$$
y^{(k+1)}(0; y_0) = P_k(f)_{t=0, y=y_0}
$$

where $P_k(f)$ is a polynomial of f and its partial dervatives up to k^{th} order evaluated at $t = 0$ and $y = y_0$. Since $P_k(f)$ has positive cofficients and by (1.19) we obtain

$$
\begin{aligned}
&\left|y^{(k+1)}(0; y_0)\right| = \left|P_k(f)\right|_{t=0, y=y_0} \leq P_k(|f|)|_{t=0, y=y_0} \\
&\leq P_k(1 + |f|)|_{t=0, y=y_0} \leq P_k(g)|_{\rho=0} = \rho^{(k+1)}(0), k \geq 0
\end{aligned}
\tag{1.20}
$$

where the last identity follows from the ODE $\frac{d\rho}{dt} = g(\rho)$. Thus, the statement of the theorem is proved. $\square$

We can now state the main result of this chapter.

Theorem 1.3.4. *Let $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ be a real analytic function in the domain $[-t_0, t_0] \times B_{\delta_0}(y_0)$ for some $t_0 > 0$ and $\delta_0 > 0$. Let $y_n(t)$ be defined by the recurrence equation (1.6). There exist a $\tau \in [0, t_0]$ such that the n^{th} partial sum of the Adomian series (1.3) converges to the solution $y(t; y_0)$ of the Volterra equation (1.2) in $C([-\tau, \tau], \mathbb{R}^d)$.*

Remark 1.3.5. *Similarly to Theorem 1.3.1 we shall prove Theorem 1.3.4 for the simplest case of $d = 1$.*

Proof. Working with iteration of the Adomian method, we set

$$y_{k+1}(t) = \int_0^t A_k(s, y_0(s), ..., y_n(s)), k \geq 0 \tag{1.21}$$

where

$$A_k = \frac{1}{k!} \frac{d^k}{d\varepsilon^k} f\left(t, y_0 + \sum_{m=1}^{\infty} \varepsilon^m y_m\right)|_{\varepsilon=0}$$

For instance, we obtain at $k = 0$

$$|y_1(t)| \leq \int_0^t |f(s, y_0)|\, ds \leq g(0)t \equiv \rho'(0)t$$

at $k = 1$

$$|y_2(t)| \leq \int_0^t |f'(s, y_0)|\, |y_1(s)|\, ds \leq \frac{t^2}{2} g'(0)g(0) \equiv \frac{t^2}{2}\rho''(0)$$

Let the following relation be true at $k = n$

$$|y_n(t)| \leq \frac{1}{n!} t^n \rho^{(n)}(0).$$

We shall prove that the same relation is true at $k = n + 1$:

$$|y_{n+1}(t)| \leq \frac{1}{(n+1)!} t^{n+1} \rho^{(n+1)}(0).$$

Let

$$Y_n(t) = \sum_{m=0}^{n} \varepsilon^m y_m(t),$$

where $\varepsilon > 0$ is a formal parameter. Then,

$$|Y_n(t)| \leq \sum_{m=0}^{n} \varepsilon^m |y_m(t)| \leq \sum_{m=0}^{n} \frac{\varepsilon^m t^m \rho^{(m)}(0)}{m!} = \left(\sum_{m=0}^{\infty} - \sum_{m=n+1}^{\infty} \right) \frac{\varepsilon^m t^m \rho^{(m)}(0)}{m!}$$

Let $m - (n+1) = l$ then

$$|Y_n(t)| \leq \rho(\varepsilon t) - \varepsilon^{n+1} t^{n+1} \sum_{l=0}^{\infty} \frac{\varepsilon^l t^l}{(1+l+n)!} \rho^{1+l+n}(0).$$

Therefore, there exist a C^∞ function $\tilde{Y}_n(t)$ on $[-\tau, \tau]$ such that

$$Y_n(t) = \rho(\varepsilon t) - \varepsilon^{n+1} t^{n+1} \tilde{Y}_n(t), \quad \forall t \in [-\tau, \tau]$$

where τ is defined by Theorem 1.3.1. The first few estimates of Adomian polynomials are given by

$$\begin{aligned}
|A_0| &\leq C = g(0) = \rho'(0) \\
|A_1| &\leq |f'|\,|y_1| \leq \frac{C}{a} g(0)t = tg(0)g'(0) = t\rho''(0) \\
|A_2| &\leq |f'|\,|y_2| + \frac{1}{2}|f''|\,|y_1^2| \leq \frac{C}{a}|y_2| + \frac{C}{a^2}|y_1|^2 \\
&\leq \frac{t^2}{2}\left(g(0)(g'(0))^2 + g''(0)(g(0))^2\right) = \frac{t^2}{2}\rho'''(0).
\end{aligned}$$

To estimate $A_n(t)$ in general case, we use formula

$$A_n(t) = \frac{1}{n!}\frac{d^n}{d\varepsilon^n} f(t, Y_n(t))|_{\varepsilon=0}$$

and compute

$$\begin{aligned}
|A_n(t)| &\leq \frac{1}{n!}\left|\frac{d^n}{d\varepsilon^n} f(t, Y_n)\right|_{\varepsilon=0} \leq \frac{t^n}{n!}\left|\frac{d^n}{d\mu^n} f(t, \rho(\mu))\right|_{\mu=\varepsilon t=0} \\
&\leq \frac{t^n}{n!}|P_n(\rho(0))| \leq \frac{t^n}{n!}\rho^{(n+1)}(0),
\end{aligned} \tag{1.22}$$

where the last inequality is obtained in (1.20). Using the iterative formula (1.21), we finally obtain

$$|y_{n+1}(t)| \leq \frac{1}{(n+1)!} t^{n+1} \rho^{(n+1)}(0).$$

Therefore, the Adomian series is majorant by the same power series as the analytic solution in Theorem 1.3.1 is. By the Weierstrass M-test, the Adomian series converges. Moreover, as follows from (1.21) the series (1.4) for Adomian polynomials converges too, so that the Adomian series solves the same Volterra integral equation (1.2) in $C\left([-\tau,\tau],\mathbb{R}\right)$. By uniqueness of solutions, the Adomian series is equivalent to the solution $y(t;y_0)$ of the Volterra equation (1.2). $\square$

1.4 Rate of convergence

In this section, a simple method to determine the rate of convergence of the ADM is introduced. Using this method, we give a bound for the error of the Adomian decomposition series.

Theorem 1.4.1. *Under the same condition as in Theorem 1.3.4, the rate of convergence is exponential in the sense that there exists $C_0 > 0$ such that*

$$E_n \le C_0 \left(\frac{2C\tau}{a}\right)^{n+1}, \quad n \ge 1$$

for all $\tau < \frac{a}{2C}$, where

$$E_n = \left\| y - \sum_{m=0}^{n} y_m \right\|,$$

and (a, C) are defined in Cauchy estimates (1.17)-(1.18).

Proof. By Theorem 1.3.4, we have

$$|y_{n+1}(t)| \le \frac{t^{n+1}\rho^{(n+1)}(0)}{(n+1)!}, \quad \forall t \in [0,\tau],$$

so that

$$\|y_{n+1}\| \le \frac{\tau^{n+1}\rho^{(n+1)}(0)}{(n+1)!},$$

where the norm $\|\cdot\|$ in $C([-\tau,\tau],\mathbb{R}^d)$ is defined by (1.9). Since $\rho(t)$ is explicitly given by

$$\rho(t) = a - \sqrt{a^2 - 2Cat},$$

then

$$\rho^{(n)}(0) = \frac{(2n-3)!!C^n}{a^{n-1}}. \tag{1.23}$$

By Theorem 1.3.4, the Adomian series $y(t) = \sum_{m=0}^{\infty} y_m(t)$ converges and the error is defined and estimated by

$$E_n = \left\| \sum_{j=n+1}^{\infty} y_j \right\| \le \sum_{j=n+1}^{\infty} \|y_j\| \le \sum_{j=n+1}^{\infty} \frac{\tau^j \rho^{(j)}(0)}{j!} \le \sum_{j=n+1}^{\infty} \frac{a}{j!} \left(\frac{C\tau}{a}\right)^j (2j-3)!!$$

Let $k = j - (n+1)$, then

$$E_n \le a \left(\frac{2C\tau}{a}\right)^{n+1} \sum_{k=0}^{\infty} \frac{(2k+2n-1)!!}{2^{k+n+1}(k+n+1)!} \left(\frac{2C\tau}{a}\right)^k .$$

Since

$$\frac{(2k+2n-1)!!}{2^{k+n+1}(k+n+1)!} \le \frac{1}{2n+2k} \le 1, \forall n \ge 1, k \ge 1,$$

we obtain

$$E_n \le a \left(\frac{2C\tau}{a}\right)^{n+1} \sum_{k=0}^{\infty} \left(\frac{2C\tau}{a}\right)^k = \frac{a\left(\frac{2C\tau}{a}\right)^{n+1}}{1 - \frac{2C\tau}{a}},$$

for all $\tau < \dfrac{a}{2C}$. The theorem is proved with $C_0 = \frac{a}{1-\frac{2C\tau}{a}}$. □

1.5 Is the Adomian iterative method related to a contraction operator?

We recall that the Picard iterative method (1.7) is related to a contraction operator provided the time interval $[-t_0, t_0]$ is small enough. We shall ask if the Adomian iteration formula (1.6) is related to a contraction operator. The question can be formulated as follows. Let $Y_n(t) = \sum_{m=0}^{n} y_m(t)$. Is there a constant $Q < 1$ such that

$$\|Y_{n+1} - Y_n\| \leqslant Q \, \|Y_n - Y_{n-1}\|, \quad \forall n \geq 1, \tag{1.24}$$

or, equivalently, $\|y_{n+1}\| \leq Q\, \|y_n\|$?. We will show, however, that the answer is negative in general. To be more precise, we will construct a counter-example for $d = 1$, which shows that no $Q < 1$ exists in a general case.

In particular, consider the first-order differential equation

$$\begin{cases} \dfrac{dy}{dt} = 2y - y^2 \\ \quad y(0) = 1 \end{cases} \tag{1.25}$$

with exact solution $y = 1 + \tanh(t)$. By the ADM, we write the above initial-value problem in the integral form:

$$y(t) = 1 + \int_0^t (2y(s) - y^2(s))ds$$

and compute the Adomian polynomials for $f(y) = 2y - y^2$ in the form

$$\begin{aligned} A_0 &= 2y_0 - y_0^2, \\ A_1 &= 2y_1 - 2y_0 y_1, \\ A_2 &= 2y_2 - 2y_0 y_2 - y_1^2, \\ A_3 &= 2y_3 - 2(y_0 y_3 + y_1 y_2), \\ A_4 &= 2y_4 - 2(y_0 y_4 + y_1 y_3) - y_2^2. \end{aligned}$$

Using (1.6), we determine few first terms of the Adomian series

$$y_0 = 1; \quad y_1 = t; \quad y_2 = 0; \quad y_3 = -\frac{t^3}{3}; \quad y_4 = 0; \quad y_5 = \frac{2t^5}{15}.$$

Therefore, Adomian iterations are not related to a contraction operator since even-numbered corrections of $y_n(t)$ are zero.

On the other hand, using Picard iterations,

$$y^{(n+1)} = 1 + \int_0^t \left(2y^{(n)}(s) - (y^{(n)}(s))^2\right) ds,$$

we obtain successive approximations in the form:

$$\begin{aligned} y^{(0)} &= 1; \\ y^{(1)} &= 1 + t; \\ y^{(2)} &= 1 + t - \frac{t^3}{3}; \\ y^{(3)} &= 1 + t - \frac{t^3}{3} + \frac{2t^5}{15} - \frac{t^7}{63}, \end{aligned}$$

and the successive approximations satisfy

$$\left\|y^{(n+1)} - y^{(n)}\right\| \leq Q \left\|y^{(n)} - y^{(n-1)}\right\|$$

for some $Q < 1$ provided that $[-t_0, t_0]$ is sufficiently small.

Note again, similarly to Section 1.2 that the Picard method mixes up powers of the partial sum for the exact solution $y(t) = 1 + \tanh(t)$, while the Adomian series is equivalent to the power series in time. Therefore, the ADM is expected to converge faster than the Picard method. We shall illustrate this point with more examples in Chapter 2.

Chapter 2

Numerical implementation of the ADM for ODEs

In this chapter we describe how to implement the ADM numerically. We also compare the ADM with the Picard and Runge-Kutta methods using MATLAB.

2.1 Numerical algorithm for the ADM and Picard method

Consider the following initial-value problem for a system of differential equations:

$$\begin{cases} \dfrac{dy}{dt} = f(t,y) \\ \quad y(0) = y_0 \end{cases} \tag{2.1}$$

where $y \in \mathbb{R}^d$, and $f : \mathbb{R} \times \mathbb{R}^d \longrightarrow \mathbb{R}^d$. For instance, if $x : \mathbb{R} \to \mathbb{R}$ satisfies the initial-value problem for the second-order differential equation

$$x'' = F(t,x,x'), \quad x(0) = A, \quad x'(0) = B, \tag{2.2}$$

then the vector $y \in \mathbb{R}^2$ with components $y_1 = x$, $y_2 = x'$ satisfies the system (2.1) with

$$f = \begin{pmatrix} y_2 \\ F(t, y_1, y_2) \end{pmatrix}, \quad y_0 = \begin{pmatrix} A \\ B \end{pmatrix} \tag{2.3}$$

Adomian Decomposition Method (ADM)

To solve equation (2.1) using the Adomian decomposition method numerically we define elements of the Adomian series by recursive equation (1.6) and apply the trapezoidal rule on $[0, T]$ with grid points at

$$t_m = mh, \quad m = 0, 1, 2, ...M,$$

where $h = \frac{T}{M}$. Then,

$$\begin{aligned} y_{n+1}(t_m) &= \frac{h}{2}(A_n(0, y_0(0), ..., y_n(0)) + A_n(t_m, y_0(t_m), ..., y_n(t_m)) \\ &+ 2\sum_{j=1}^{m-1} A_n(t_j, y_0(t_j), ..., y_n(t_j))) \end{aligned} \tag{2.4}$$

where $y_0(t) = y_0$ and $y_n(0) = 0$ for $n \geq 1$.

After Adomian polynomials A_n are computed recursively in the explicit form for $n = 0, 1, 2, ..., N$, we can use the trapezoidal rule (2.4) on the grid $\{t_m\}_{m=0}^{M}$ by incrementing n from $n = 0$ to $n = N$. Thus, we can define the n^{th}-partial sum of the Adomian series on the grid $\{t_m\}_{m=0}^{M}$ by

$$Y_n(t_m) = y_0 + \sum_{i=1}^{n} y_i(t_m)$$

for $n = 1, 2, ..., N$ and $m = 1, 2, ..., M$.

Picard Method (PM)

To solve equation (2.1) using the Picard method numerically we take the recursive equation (1.7) and apply trapezoidal rule on the same grid in the form:

$$y^{(n+1)}(t_m) = y_0 + \frac{h}{2}\left(f(0, y^{(n)}(0)) + f(t_m, y^{(n)}(t_m)) + \sum_{j=1}^{m-1} f(t_j, y^{(n)}(t_j))\right) \tag{2.5}$$

where $y^{(0)}(t) = y_0$ and $y^{(n)}(0) = y_0$ for all $n \geq 1$.

Runge-Kutta Method (RKM)

To solve equation (2.1), using the Runge-Kutta method, we take the standard Runge-Kutta method of the fourth-order given by

$$\begin{aligned} y_{m+1} &= y_m + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \\ t_{m+1} &= t_m + h, \end{aligned} \tag{2.6}$$

where y_m is a numerical approximation of the solution $y(t; y_0)$ at $t = t_m$ and

$$\begin{aligned} k_1 &= f\left(t_m, y_m\right), \\ k_2 &= f\left(t_m + \frac{h}{2}, y_m + \frac{h}{2}k_1\right), \\ k_3 &= f\left(t_m + \frac{h}{2}, y_m + \frac{h}{2}k_2\right), \\ k_4 &= f\left(t_m + h, y_m + hk_3\right). \end{aligned} \tag{2.7}$$

2.2 Two numerical examples

Two examples of the initial-value problem for second-order differential equations are considered here. In the first example, we compare the Adomian decomposition method and the Runge-Kutta method. In the second example, we compare the Adomian decomposition method and the Picard method. The numerical computations are performed using MATLAB.

Example 2.1

Let us consider the nonlinear differential equation:

$$-\frac{d^2y}{dt^2} + \left(1 - \frac{3}{\cosh^2(t)}\right) y(t) + y^3(t) = 0 \tag{2.8}$$

with initial conditions $y(0) = 1$ and $y'(0) = 0$. The exact solution for this initial-value problem is $y_{exact}(t) = sech(t)$.

Equation (2.8) is a stationary Gross-Pitaevskii equation that describes, for example, localization of an atomic gas in trapped Bose-Einstein condensates.

Approximation by the ADM

We first compute the Adomian polynomials for $f(y) = y^3$ using generating rule (1.5). The first four polynomials are

$$\begin{aligned} A_0 &= y_0^3, \\ A_1 &= 3y_0^2 y_1, \\ A_2 &= 3y_0 y_1^2 + 3y_0^2 y_2, \\ A_3 &= 3y_3 y_0^2 + 6y_2 y_1 y_0 + y_1^3. \end{aligned}$$

A general formula is also available:

$$A_k = \sum_{i=0}^{k} \sum_{j=0}^{k-i} y_i y_j y_{k-i-j}$$

Integrating twice the differential equation, we obtain the recursive formula for the ADM in the form:

$$y_{n+1}(t) = \int_0^t d\tau \int_0^\tau \left(\left[1 - \frac{3}{\cosh^2(s)}\right] y_n(s) + A_n(y_0(s), ..., y_n(s)) \right) ds, \quad n \geq 0$$

starting with $y_0 = 1$. If $Y_n(t)$ is a partial sum of the Adomian series, the approximation error of the ADM is defined by

$$E_n^{ADM}(T) = \|Y_n - y_{exact}\| = \sup_{t \in [0,T]} |Y_n(t) - y_{exact}(t)| .$$

T	Error RK	Error AD $n = 15$	Error AD $n = 30$
0.5	0.7131×10^{-10}	0.8103×10^{-10}	0.7128×10^{-14}
1	0.1633×10^{-9}	0.6215×10^{-8}	0.8754×10^{-13}
1.5	0.3123×10^{-9}	0.7588×10^{-6}	0.5421×10^{-11}
2	0.4907×10^{-9}	0.8752×10^{-4}	0.2112×10^{-9}

Table 2.1: Comparison of errors between the ADM and the RKM

The error is evaluated on the discrete set $\{t_m\}_{m=0}^{M}$ for a numerical approximation of $Y_n(t)$.

Approximation by the RKM

Let $y = x_1$, $y' = x_2$, and write equation (2.8) in the form

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= \left(1 - \frac{3}{\cosh^2 t}\right) x_1 + x_1^3 \end{aligned}$$

Runge-Kutta method computes the approximations by using (2.6) and (2.7) for (x_1, x_2). The approximation error of the Runge-Kutta method is defined by:

$$E^{RKM}(T) = \|y_{RK} - y_{exact}\|,$$

where y_{RK} is the numerical approximation obtained on the discrete grid $\{t_m\}_{m=0}^{M}$.

Table 2.1 shows comparison of the errors between the two methods. We find that the approximation obtained from the Adomian method with $n = 15$ is less accurate than the approximation obtained from the Runge-Kutta method for $T \geq 0.5$. On the other hand, the Adomian method with $n = 30$ gives a smaller approximation error than the Runge-Kutta method for all $T \leq 2$. Therefore, the ADM is superior to the Runge-Kutta method for smaller time intervals (for which we proved convergence of the Adomian series in Chapter 1) but the Runge-Kutta method might be more accurate for longer time intervals.

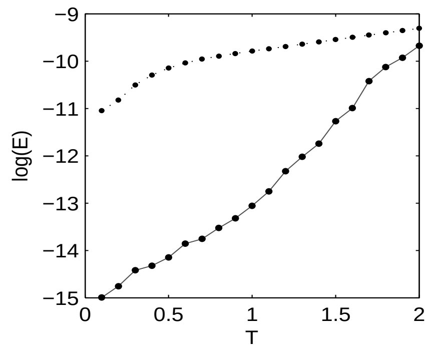

Figure 2.1: Comparison of errors between the ADM (solid curve) with $n = 30$ and the RKM (dotted curve) for $T = 2$

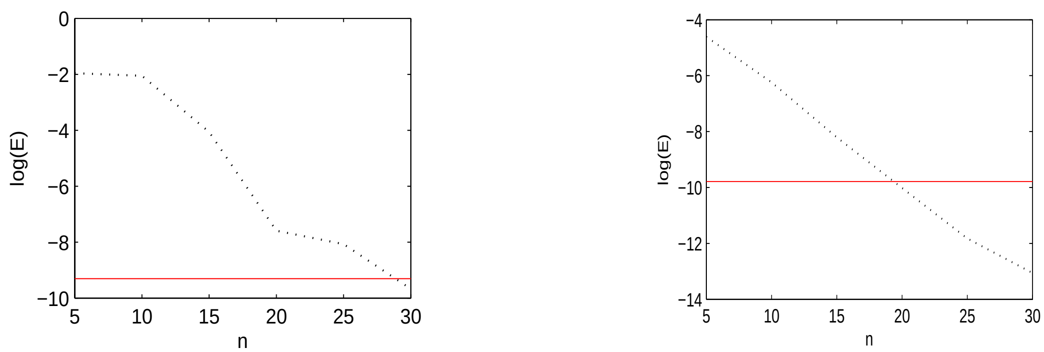

Figure 2.2: Graph of the approximation error of the ADM (dotted curve) versus n and the approximation error of the RKM (solid curve) for $T = 1$ (right) and $T = 2$ (left).

Figure 2.1 shows that the error of the RKM increases much slower than the error of the ADM with a fixed $n = 30$. If n is fixed, there exists a value of $T = T_0$ such that the error of the RKM is smaller than that of the ADM for $T > T_0$.

This tendency is also seen on Figure 2.2 for $T = 1$ (left) and $T = 2$ (right), where the errors are plotted versus n. For a given T, there exists a value of $n = n_0$ such that the error of the ADM is smaller than that of the RKM for $n > n_0$.

Example 2.2

Let us consider the nonlinear differential equation:

$$\frac{d^2y}{dt^2} + e^{-2t}y^3 = 2e^t \tag{2.9}$$

subject to the initial conditions, $y(0) = y'(0) = 1$, which has exact solution $y(t) = e^t$. This example was previously solved by El-Kalla [20]. He introduced a new definition of the Adomian polynomials:

$$\begin{aligned} \tilde{A}_0 &= f(t, y_0), \\ \tilde{A}_n &= f(t, Y_n) - f(t, Y_{n-1}), \end{aligned}$$

so that

$$\sum_{i=0}^{n} \tilde{A}_i(t, y_0, ..., y_i) = f(t, Y_n(t)).$$

He used the ODE (2.9) to claim that the Adomian series solution using the new definition of $\tilde{A}_n$ converges faster than the one constructed using the old definition of A_n. However, the new formula is nothing but the Picard iteration formula since

$$\begin{aligned} Y_{n+1} &= y_0 + \sum_{i=0}^{n} \int_0^t \tilde{A}_i(s, y_0(s), ...y_i(s))ds \\ &= y_0 + \int_0^t f(s, Y_n(s))ds. \end{aligned}$$

Therefore, we can use this example to compare the ADM and the PM as well as to check the claim of El-Kalla [20].

Approximation by the ADM

Integrating twice the differential equation (2.9), we obtain the integral equation

$$y(t) = 2e^t - t - 1 - \int_0^t d\tau \int_0^\tau e^{-2s}y^3(s)ds.$$

Using the same Adomian polynomials for $f(y) = y^3$ as in the previous example, we define

$$y_0(t) = 2e^t - t - 1$$

and compute $y_{k+1}(t)$ for $k \geq 0$ by using

$$y_{k+1}(t) = -\int_0^t d\tau \int_0^\tau e^{-2s} A_k(s) ds, k \geqslant 0.$$

The relative approximation error of the ADM is defined by:

$$RAE_n^{ADM} = \frac{||Y_n - y_{exact}||}{||y_{exact}||}.$$

Approximation by PM

By using the Picard iterative formula (1.7), we have

$$y^{(n+1)}(t) = 2e^t - t - 1 - \int_0^t d\tau \int_0^\tau e^{-2s} (y^{(n)}(s))^3 ds, \quad n \geq 0$$

starting with

$$y^{(0)}(t) = 2e^t - t - 1 \equiv y_0(t).$$

The relative approximation error of the PM is defined by:

$$RAE_n^{PM} = \frac{||y^{(n)} - y_{exact}||}{||y_{exact}||}.$$

Table 2.2 (left) demonstrates the relative approximation error of the two methods for $n = 3$ where the approximations for Y_3 and $y^{(3)}$ have been computed analytically.

Table 2.2 (right) shows the relative approximation error of the two methods for $n = 7$ where the approximations for Y_7 and $y^{(7)}$ have been computed numerically.

From the two tables we conclude that the ADM is more accurate than the PM for all the time intervals.

Figures 2.3 compares relative errors in the ADM and the PM. We note that solution using the Adomian formula converges faster than the solutions using the Picard method in contradiction to the claim of [20]. We think that the wrong claim of [20] was made due to mis-calculation of Adomian polynomials.

T	RAE_3^{ADM}	RAE_3^{PM}
0.5	3.6842×10^{-4}	4.1237×10^{-4}
1	1.4792×10^{-2}	2.693488×10^{-2}
1.5	1.358×10^{-1}	3.450×10^{-1}
2	5.786×10^{-1}	2.1437

T	RAE_7^{ADM}	RAE_7^{PM}
0.5	1.7579×10^{-4}	9.021×10^{-2}
1	2.2614×10^{-4}	6.989×10^{-1}
1.5	6.521×10^{-3}	1.9599
2	4.455×10^{-1}	20.4631

Table 2.2: Comparison between ADM and PM using analytical computations at $n = 3$ (left). Comparison between ADM and PM using numerical computations at $n = 7$ (right)

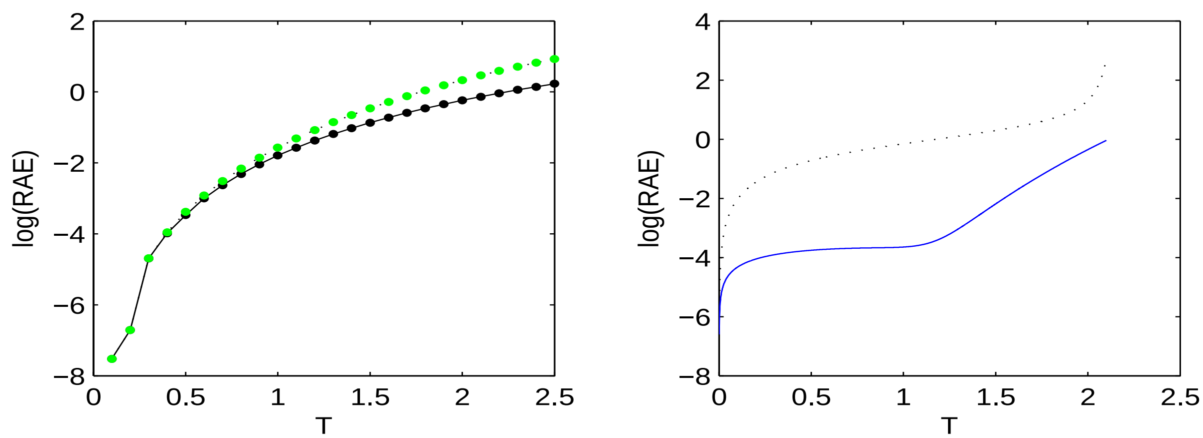

Figure 2.3: Graph of RAE_n^{ADM} (solid curve) and RAE_n^{PM} (dotted curve) using analytical computations at $n = 3$ (left) and numerical computations at $n = 7$ (right).

Chapter 3

Convergence of the ADM for PDEs

In this chapter we analyze convergence of the ADM for nonlinear partial differential equations in the form

$$u_t = Lu + N(u); \tag{3.1}$$

where L is an unbounded differential operator from a Banach space X to a Banach space Y, $(X \subseteq Y)$, and $N(u)$ is a nonlinear function that maps an element of X to an element of X.

For example, we can consider a nonlinear Schrödinger equation (NLS) in the form

$$iu_t = -u_{xx} + V(x)u + |u|^2 u; \tag{3.2}$$

where $i = \sqrt{-1}$, $V(x)$ is an external potential for $x \in \mathbb{R}$, and $u = u(x,t)$ is a complex valued function. The NLS equation plays an important role in the modeling of several physical phenomena such as the propagation of optical pulses, waves in fluids and plasma, self-focusing effects in lasers, and trapping of atomic gas in Bose-Einstein condensates.

The NLS equation (3.2) is a particular example of the general PDE (3.1) where $L = i\partial_x^2$, $N(u,x) = -i\left(V(x) + |u|^2\right)u$ and the Banach spaces are $X = H^s(\mathbb{R})$ and

$Y = H^{s-2}(\mathbb{R})$ for any $s > \frac{1}{2}$ (assuming that $V \in H^s$). The initial-value problem for the PDE (3.1) can be set from the initial data

$$u(x,0) = f(x), \quad \forall x \in \mathbb{R} \tag{3.3}$$

where $f(x) \in H^s(\mathbb{R})$ for any $s > \frac{1}{2}$.

3.1 Convergence Analysis

Let $E(t)$ be a fundamental solution operator associated with the linear Cauchy problem

$$\begin{cases} v_t = Lv \\ v(0) = f \in X \end{cases} \tag{3.4}$$

so that $v(t) = E(t)f$. For symbolic notations, we write $E(t) = e^{tL}$. In what follows, we shall assume that

$$\|E(t)f\|_X \leq C\,\|f\|_X \tag{3.5}$$

For instance if $L = i\partial_x^2$, then the initial-value problem the linear Schrödinger equation (3.4) is solved in the Fourier transform form as

$$v(x,t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-i\xi^2 t + i\xi x} \widehat{f}(\xi) d\xi, \quad \forall (x,t) \in \mathbb{R}^2,$$

where

$$\widehat{f}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-i\xi x} f(x) dx, \forall \xi \in \mathbb{R}.$$

Therefore, $E(t)$ is defined in the Fourier transform form by $\widehat{E}(t) = e^{-i\xi^2 t}$. By Parseval's identity, $E(t)$ preserves the H^s- norm in the sense that

$$\begin{aligned} \|E(t)f\|_{H^s}^2 &= \frac{1}{2\pi} \int_{\mathbb{R}} (1+\xi^2)^s \left|\widehat{E(t)f}\right|^2 d\xi \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} (1+\xi^2)^s \left|\widehat{E(t)}\right|^2 \left|\widehat{f}\right|^2 d\xi \\ &= \frac{1}{2\pi} \int_{\mathbb{R}} (1+\xi^2)^s \left|\widehat{f}\right|^2 d\xi \\ &= \|f\|_{H^s}^2, \end{aligned}$$

so that the assumption (3.5) holds with $C = 1$.

By Duhamel's principle, the initial-value problem (3.1) can be reformulated as an integral equation

$$u(t) = E(t)f + \int_0^t E(t-s)N(u(s))ds \tag{3.6}$$

Remark 3.1.1. *If $L : X \to Y$, $N : X \to X$, and $\|E(t)f\|_X \leq C\|f\|_X$ for some $C > 0$, then there exists a unique fixed-point of the integral equation (3.6) in space $C([0,T],X)$ for a sufficiently small $T > 0$, which corresponds to a unique solution of the PDE problem (3.1) in space $u(t) \in C([0,T],X) \bigcap C^1([0,T],Y)$.*

To set up the Adomian method, define

$$u(t) = \sum_{n=0}^{\infty} u_n(t) \tag{3.7}$$

where $u_0(t) = E(t)f$ and

$$u_{n+1}(t) = \int_0^t E(t-s)A_n(u_0(s), ...u_n(s))ds, \quad n \geq 0, \tag{3.8}$$

where A_n is the same Adomian polynomial as in Chapter 1 generated from an analytic function $N(u)$.

We would like to prove convergence of the Adomian series (3.7) in space X.

Theorem 3.1.2. *Let $N : X \to X$ be a real analytic function in the ball $B_a(f) \subset X$ for some radius $a > 0$. Let $L : X \to Y$ satisfy $\|E(t)f\|_X \leq C\|f\|_X$ for some $C > 0$. Let $u_0(t) = E(t)f$ and $u_n(t)$ for $n \geq 1$ be defined by the recurrence equation (3.8). There exist a $T > 0$ such that the n^{th} partial sum of the Adomian series (3.7) converges to the solution u of the equation (3.6) in $C([0,T],X)$.*

Proof. Assume that $N(u)$ is analytic in $u \in X$. Then, by Cauchy estimates, there exist $a > 0$, and $b > 0$ such that

$$\left\|\partial_u^k N(f)\right\|_X \leq \frac{bk!}{a^k}, \quad k \geq 0. \tag{3.9}$$

The Taylor series for $N(u)$ at $u = f$

$$N(u) = \sum_{k=0}^{\infty} \frac{1}{k!} \left[\partial_u^k N(f)\right] (u - f)^k,$$

converges for any $\|u - f\|_X < a$, and moreover, we obtain that

$$\begin{aligned}
\|N(u)\|_X &\leq \sum_{k=0}^{\infty} \frac{1}{k!} \frac{bk!}{a^k} \|u - f\|_X^k \\
&\leq \frac{b}{1 - \frac{\|u-f\|_X}{a}} \\
&= \frac{ba}{a - \rho} \equiv g(\rho)
\end{aligned}$$

where $\rho = \|u - f\|_X < a$. It is now clear that $\left\|\partial_u^k N(f)\right\|_X \leq g^{(k)}(0)$ for any $k \geq 0$.

Working with equation (3.8), we find that $\|u_0 - f\|_X \leq (C + 1) \|f\|_X \equiv a$, and

$$\begin{aligned}
\|u_1\|_X &\leq \int_0^t \|E(t-s)A_0\|_X \, ds \leq C \int_0^t \|A_0\|_X \, ds \\
&\leq Cg(0)t = Ct\rho'(0), \\
\|u_2\|_X &\leq \int_0^t \|E(t-s)A_1\|_X \, ds \leq C \int_0^t \|A_1\|_X \, ds \\
&\leq C^2 g'(0)g(0)t = C^2 \frac{t^2}{2} \rho''(0),
\end{aligned}$$

proving by induction that

$$\|u_{n+1}(t)\|_X \leq \frac{C^{n+1}}{(n+1)!} t^{n+1} \rho^{(n+1)}(0).$$

Therefore, the Adomian series in X is majorant by the convergent power series for $\rho(t) = a - \sqrt{a^2 - 2abCt}$ for any $t \in [0, T]$ for $T < \frac{a}{2bC}$, in full correspondence with the proof of Theorem 1.3.4. $\square$

3.2 Numerical examples

Consider the nonlinear Schrödinger equation in the form

$$\begin{cases} iu_t = -u_{xx} - 3(sech(x))^2 u + |u|^2 u \\ u(x,0) = f(x) \end{cases} \tag{3.10}$$

We shall consider (3.10) on the interval $x \in [-L, L]$ subject to periodic boundary conditions.

To find u_0, we approximate numerically the solution of equation (3.4). Using trigonometric approximation [23] on the symmetric interval $[-L, L]$, and periodic continuation to the interval $[0, 2L]$, the function $f(x)$ is interpolated at the discrete grid $\{x_k\}_{k=0}^{n-1} \in [0, 2L]$, by the trigonometric sum

$$f_k = \frac{1}{n}\sum_{j=0}^{n-1} c_j e^{\frac{2\pi jik}{n}}, \quad k = 0, 1, ..., n-1,$$

where n is even, the grid points are given by

$$x_k = \frac{2Lk}{n} \quad k = 0, 1, ..., n-1,$$

and the continuation of $f(x)$ from $[-L, 0]$ to $[L, 2L]$ is defined by

$$f(2L - x) = f(-x), \quad \forall x \in [0, L].$$

The discrete Fourier transform is defined by

$$c_j = \sum_{k=0}^{n} f_k e^{\frac{-2\pi jik}{n}}, \quad j = 0, 1, ..., n-1.$$

where c_0 and $c_{\frac{n}{2}}$ are real, and

$$c_{-j} = \overline{c}_{n-j}, \quad j = 0, 1, ..., \frac{n}{2}.$$

The function $u(x,t)$ can be approxmated at any time instances $\{t_m\}_{m=0}^{M}$ on the interval $[0,T]$ by applying the inverse discrete Fourier transform to $\left\{e^{-it_m\xi_j^2}c_j\right\}_{j=0}^{n-1}$, where

$$\xi_j = \begin{cases} \frac{\pi j}{L}, & j = 0, 1, ..., \frac{n}{2} - 1 \\ \frac{-\pi(n-j)}{L}, & j = \frac{n}{2}, ..., n-1 \end{cases}$$

Example 3.1

Consider the nonlinear Schrödinger equation (3.10) with $f(x) = sech(x)$. The exact solution of the initial-value problem (3.10) is $U_{exact} = e^{it}sech(x)$.

First, we compute the Adomian polynomials for $N(u) = |u|^2\, u$ using the explicit formula

$$A_k = \sum_{i=0}^{k}\sum_{j=0}^{k-i} \overline{u}_i u_j u_{k-i-j}.$$

Integrating equation (3.10), we obtain the recursive formula for the ADM in the form:

$$u_{n+1}(x,t) = 3i\int_0^t E(t-s)(sech(x))^2 u_n(x,s)ds - i\int_0^t E(t-s)A_n(u_0, ..., u_n)ds.$$

for $n \geq 0$ and $u_0 = E(t)f$, where $E(t) = e^{itL}$ and $f = sech(x)$.

To express integrals on $[0,T]$ numerically, we use a discrete grid $\{t_m\}_{m=0}^{M}$ and the trapezoidal rule similarly to the algorithm in Section 2.1.

Figure 3.1 shows the first two approximations of the ADM. The approximation u_0 decays in time, and the approximation $u_0 + u_1$ grows gradually in time.

Figure 3.2 (left) compares absolute errors E_n of the ADM for $n = 0, 1, ..., 10$, where

$$E_n(T) = \sup_{t\in[0,T]}\left(\sup_{x\in[-L,L]} |U_n - U_{exact}|\right),$$

and $U_n = u_0 + u_1 + ... + u_n$. We note that the errors E_n decrease with increasing n for any fixed t. Figure 3.2 (right) shows the approximation U_{10} that remains nearly constant in amplitude as time t evolves, similarly to the exact solution $U_{exact} = e^{it}sech(x)$.

t	E_0	E_5	E_{10}
0.2	0.3606	0.001	0.0002
0.4	0.6306	0.004	0.0007
0.6	0.8413	0.0148	0.0025
0.8	1.0113	0.0388	0.0065
1	1.1501	0.0812	0.0137

Table 3.1: Comparison of absolute errors between U_0, U_5, and U_{10}

Table 3.1 shows the absolute errors $E_n(T)$ versus T for $n = 0, 5, 10$. This table illustrates that the errors are smaller for smaller values of T and larger values of n.

Example 3.2

Consider the same nonlinear Schrödinger equation (3.10) but with initial condition $f(x) = sech^2(x)$. In this case, we can't find the exact solution but we can still approximate solutions numerically using the same MATLAB code as in Example 3.1.

Figure 3.3 shows the numerical approximations U_0, U_5, U_{10} and U_{20} on a grid for $x \in [-10, 10]$ and $t \in [0, 2]$. We can see that U_0 is decaying in time, U_5 and U_{10} increase in time, while U_{20} is the closest approximation to the actual solution, which describes a transition of the initial data to the soliton solutions of the NLS equation (3.10) and periodic oscillations of solitary waves. Increase in amplitudes of the approximations U_5 and U_{10} and visible oscillations in the approximation U_{20} near the end of computational interval at $t = 2$ can be related to the divergence of the Adomian series for large time instances.

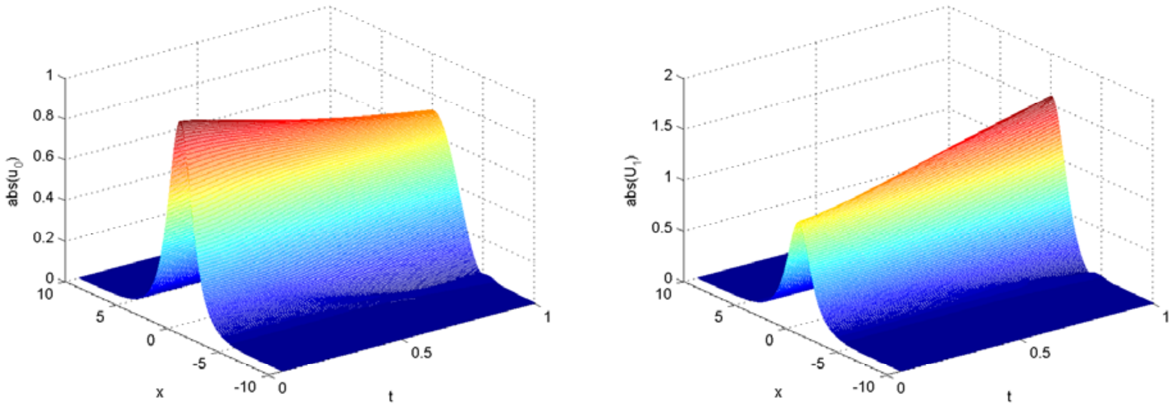

Figure 3.1: The approximations $|u_0|$ (left) and $|u_0 + u_1|$ (right), for $-10 \le x \le 10$ and $0 \le t \le 1$.

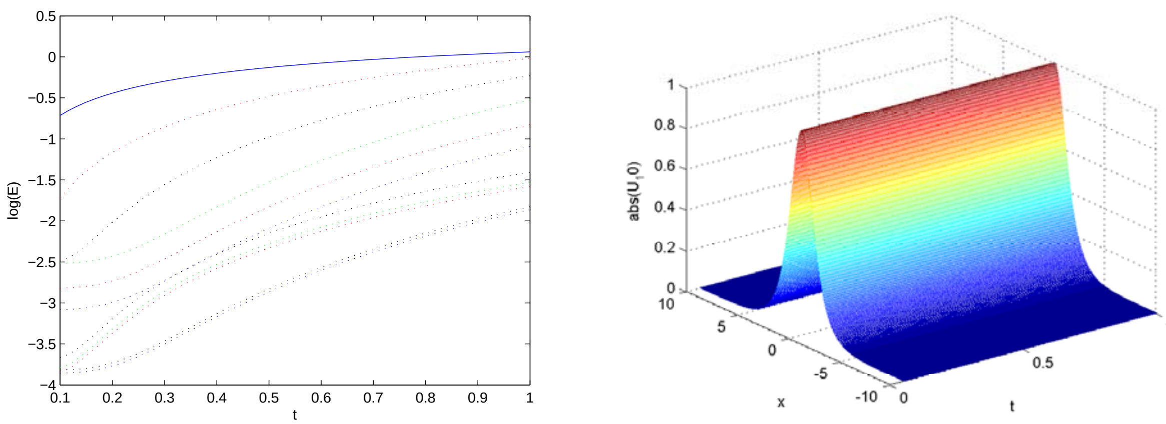

Figure 3.2: Comparison of absolute errors E_n for the ADM with $n = 0, 1, ..., 10$ (left) and the surface $|U_{10}|$ (right) for $-10 \le x \le 10$ and $0 \le t \le 1$.

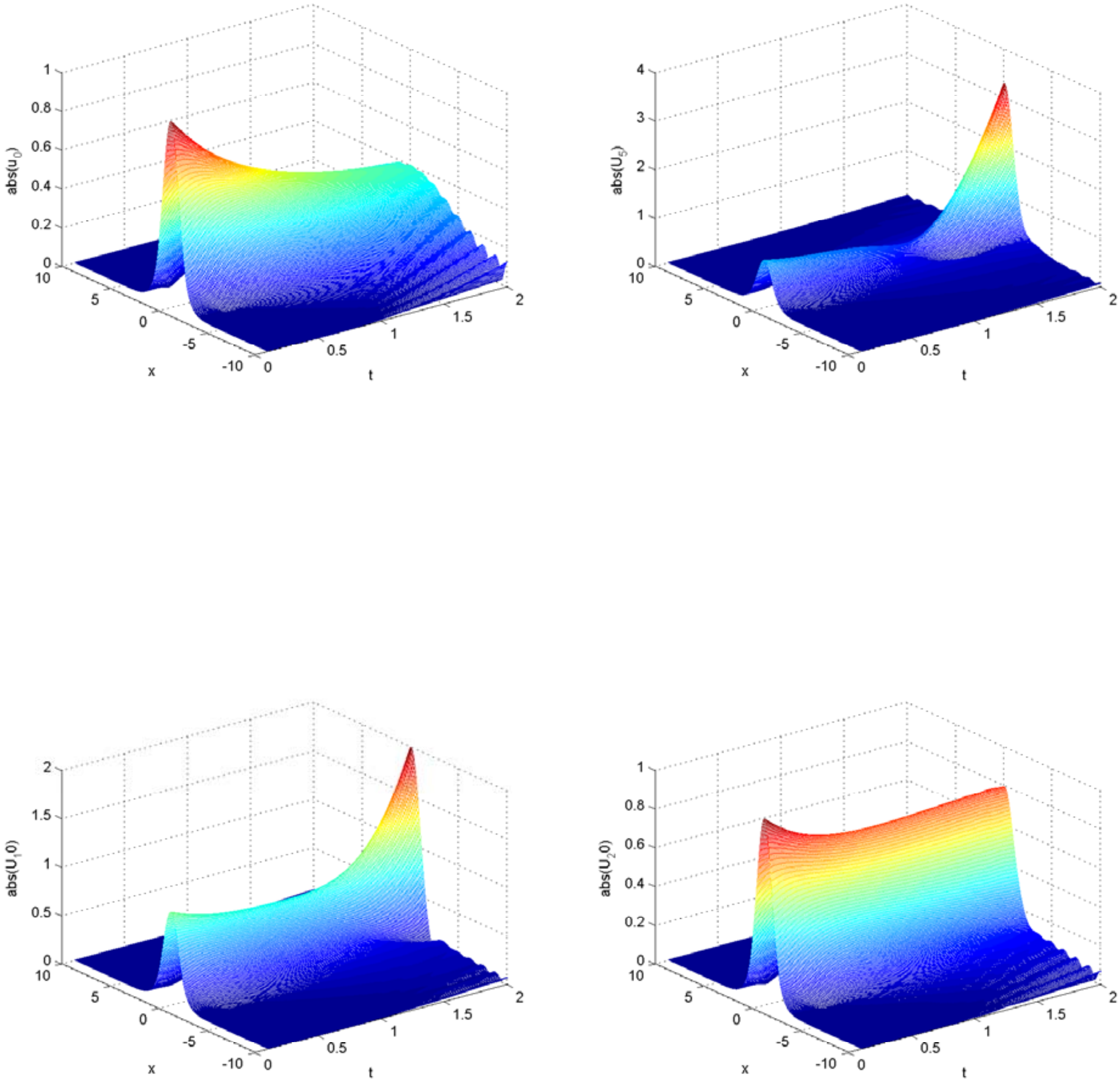

Figure 3.3: The approximations $|u_0|$ (top left), $|U_5|$ (top right), $|U_{10}|$ (bottom left) and $|U_{20}|$ (bottom right) for $-10 \leq x \leq 10$, $0 \leq t \leq 2$.

Appendix: Numerical codes

Appendix 1 Example 2.1

```
clear all
close all
y(1)=0; x(1)=1; h=0.01;
t=0:h:2; k=length(t)-1;
for n=1:k;
    m1=y(n);
    n1=((-3./(cosh(t(n))).^2)+1).*x(n)+(x(n)).^3;
    m2=y(n)+n1.*h./2;
    n2=((-3./(cosh(t(n)+h./2)).^2)+1).*(x(n)+m1.*h./2)+(x(n)+m1.*h./2).^3;
    m3=y(n)+n2.*h./2;
    n3=((-3./(cosh(t(n)+h./2)).^2)+1).*(x(n)+m2.*h./2)+(x(n)+m2.*h./2).^3;
    m4=y(n)+n3.*h;
    n4=((-3./(cosh(t(n)+h)).^2)+1).*(x(n)+m3.*h)+(x(n)+m3.*h).^3;
    x(n+1)=x(n)+h./6.*(m1+2.*m2+2.*m3+m4);
    y(n+1)=y(n)+h./6.*(n1+2.*n2+2.*n3+n4);
end
Exact = 1./cosh(t);
E=abs(Exact-x);
```

```
plot(t,log10(E),'k.')
hold on
clear all
h=0.01; s=0:h:2; d=30;n=length(s);
y=zeros(n,d+1); y(:,1)=1;y(1,(2:d))=0;Y=zeros(n,d+1);x=zeros(n,d);
for i=1:d;
    V=0;
    for k=1:i;
        Z=0;
        for j=1:i-k+1
            Z=Z+y(:,k).*y(:,j).*y(:,i-k-j+2);
        end
        V=V+Z;
    end
    x(:,i)=(((-3./(cosh(s')).^2))+1).*y(:,i)+V;
    for k1=2:n;
        Y(k1,i)=(h/2)*(x(1,i)+2.*sum(x(2:k1-1,i))+x(k1,i));
    end
    for k2=2:n
     y(k2,i+1)=(h/2)*(Y(1,i)+2.*sum(Y(2:k2-1,i))+Y(k2,i));
    end
end
```

Appendix 2 Example 2.2

```
clear all
y=0:0.1:2.5;
x0=2.*exp(y)-y-1;
```

```
x1 = (1/8.*(569.*exp(2.*y)-157.*y.*exp(2.*y)-64.*exp(3.*y)+16.*y.^3
.*exp(2.*y)+48.*y.^2.*exp(2.*y)-48.*y.^2.*exp(y)-288.*y.*exp(y)
-528.*exp(y)+2.*y.^3+12.*y.^2+27.*y+23)).*exp(-2.*y);
x2=-(1/1658880.*(4729995+816480.*y.^4+9732150.*y
+787553280.*y.^2.*exp(4.*y)+77760.*y.^5+12976492085.*exp(4.*y)
+1725261120.*exp(2.*y)-14340188160.*exp(3.*y)
+3596400.*y.^3+8242560.*y.^2-207042560.*exp(y)-159252480.*exp(5.*y)
+1555277760.*y.*exp(2.*y)+111196800.*y.^3.*exp(2.*y)
+522547200.*y.^2.*exp(2.*y)-166717440.*y.^2.*exp(y)-301854720.*y.*exp(y)
+2488320.*y.^5.*exp(2.*y)-38568960.*y.^3.*exp(4.*y)+1990656.*y.^5.*exp(4.*y)
-2276812800.*y.^2.*exp(3.*y)-8062156800.*y.*exp(3.*y)+9953280.*y.^4
.*exp(4.*y)-39813120.*y.^4.*exp(3.*y)
-3870720.*y.^4.*exp(y)-4533619530.*y.*exp(4.*y)+24883200.*y.^4.*exp(2.*y)
-40734720.*y.^3.*exp(y)-477757440.*y.^3.*exp(3.*y))).*exp(-4.*y);
x3 = 4.252672720*10^(-15)*(2.333575263*10^15.*y+
9.170703360*10^14 .*y.^7.*exp(4.*y)+3.793465961*10^18.*y.^3.*(exp(4.*y))
+1.423641875*10^19.*y.^2.*(exp(4.*y))
-4.389396480*10^15 .*y.^6.*(exp(3.*y))-1.625330811*10^17.*y.^5
.*(exp(5.*y))+1.199731405*10^19
.*y.^2.*(exp(6.*y))+3.684701270*10^19.*y.*(exp(4.*y))+1.055504660*10^18
.*y.^2.*(exp(2.*y))+1.620304560*10^15.*y.^6.*(exp(2.*y))-1.022886144*10^15
.*y.^5.*(exp(6.*y))+8.062156800*10^13.*y.^7.*(exp(6.*y))
+1.102248000*10^14.*y.^7.*(exp(2.*y))
+8.381304324*10^14+1.597683724*10^18.*y.*(exp(2.*y))
+9.788233951*10^17.*(exp(2.*y))+3.240405000*10^13.*y.^6+1.957319792*10^15.*
y.^3+2.836264942*10^15 .*y.^2+2.168519850*10^14 .*y.^5+8.301220200*10^14
.*y.^4+2.143260000*10^12 .*y.^7-1.336100936*10^18.*
y.^4.*(exp(5.*y))-1.062010196*10.^19 .*y .*(exp(3.*y))
```

```
+7.324217510*10^17 .*y.^4
.*(exp(4.*y))+2.047087915*10^20.*(exp(6.*y))-2.393044347*10^20
.*(exp(5.*y))-3.611846246*10^17.*(exp(7.*y))-8.315929843*10^18.*(exp(3.*y))
-6.928975872*10^16.*y.^5.*(exp(3.*y))+5.643509760*10^14.*y.^6.*(exp(6.*y))
-9.684262748*10^18 .*y.^3 .*(exp(5.*y))-9.029615616*10^15
.*y.^6.*(exp(5.*y))-4.597762176*10^17.* y.^4 .*(exp(3.*y))-1.946971676*10^18
.*y.^3 .*(exp(3.*y))+8.882355283*10^16 .*y.^5.*(exp(4.*y))+1.283898470*10^16
.*y.^6.*(exp(4.*y))-5.820252641*10^18 .*y.^2.*(exp(3.*y))-1.306444298*10^18
.*y.^3 .*(exp(6.*y))+7.963183022*10^16 .*y.^4.*(exp(2.*y))+4.234632786*10^19
.*(exp(4.*y))-1.330406486*10^20 .*y.*(exp(5.*y))+3.715580413*10^17.*y.^3
.*(exp(2.*y))+1.254036735*10^16 .*y.^5 .*(exp(2.*y))-4.137011597*10^19
.*y.^2.*(exp(5.*y))+1.229226970*10^17.*
y.^4.*(exp(6.*y))-5.323172094*10.^16.*(exp(y))-7.021322262*10.^19
.*y.*(exp(6.*y))-1.593115776*10.^14.*y.^6.*(exp(y))
-5.449132718*10.^16.*y.^3.*(exp(y))
-1.557727247*10.^16.*y.^4.*(exp(y))-2.410582084*10.^15.*y.^5.*(exp(y))
-1.087212350*10.^17.*y.^2.*(exp(y))
-1.172769145*10.^17.*y.*(exp(y))).*(exp(-6.*y));
x=x0+x1+x2;
xx0=2.*exp(y)-y-1;
xx1=(1/8.*(569.*exp(2.*y)-157.*y.*exp(2.*y)-64.*exp(3.*y)
+16.*y.^3.*exp(2.*y)+48.*y.^2.*exp(2.*y)-48.*y.^2.*exp(y)
-288.*y.*exp(y)-528.*exp(y)+2.*y.^3+12.*y.^2
+27.*y+23)).*exp(-2.*y);
xx=xx0+xx1+xx2;
xExact=exp(y);
Re1=abs(x-xExact)./(xExact)
Re2=abs(xx-xExact)./(xExact)
```

```
plot(y,log10(Re1),'k:')
hold on
plot(y,log10(Re2),'r:')

clear all
close all
h=0.01; s=0:h:2.1; d=8;n=length(s);
y=zeros(n,d+1); y(:,1)=2*exp(s')-s'-1;
y(1,(2:d+1))=y(1,1);
Y=zeros(n,d+1); x=zeros(n,d);
for i=1:d;
    x(:,i)=-(exp(-2*s')).*((y(:,i)).^3);
    for k1=2:n;
       Y(k1,i+1)=(h/2).*(x(1,i)+2*sum(x(2:k1-1,i))+x(k1,i));
    end
    for k2=2:n
        y(k2,i+1)=y(k2,1)+(h/2)*(Y(1,i+1)+2.*sum(Y(2:k2-1,i))+Y(k2,i+1));
    end
end
yExact=exp(s');
S=y(:,d+1);
E1=abs(S-yExact);
REAPI=E1./yExact;
clear all
h=0.01; s=0:h:2.1; d=7;n=length(s);
y=zeros(n,d+1); y(:,1)=2*exp(s')-s'-1;
y(1,(2:d+1))=0; Y=zeros(n,d+1);
Y(1,2:8)=0; Y(:,1)=2*exp(s')-1;
```

```
x=zeros(n,d);
for i=1:d;
    V=0;
    for k=1:i;
        Z=0;
        for j=1:i-k+1
            Z=Z+y(:,k).*y(:,j).*y(:,i-k-j+2);
        end
        V=V+Z;
    end
    x(:,i)=-(exp(-2*s')).*V;
    for k1=2:n;
        Y(k1,i+1)=(h/2)*(x(1,i)+2.*sum(x(2:k1-1,i))+x(k1,i));
    end
for k2=2:n
        y(k2,i+1)=(h/2)*(Y(1,i+1)+2.*sum(Y(2:k2-1,i+1))+Y(k2,i+1));
    end
 end
yExact=exp(s');
```

Appendix 3   Example 3.1

```
clear all
close all
a = 10; N = 400; m=N/2; d=11;
dx = 2*a/N;h=0.005;
t=0:h:1; M=length(t); x = -a : dx : a-dx;
j=-m:1:m-1; xi=(pi/a)*j;
```

```
u=zeros(length(t),length(x),d);
 g = sech(x);
 FF =[g(m+1:N),g(1:m)] ;
 C=fft(FF);
c=[C(m+1:N),C(1:m)];
for s=1:M
    uhat = exp(-i.*t(s).*(xi).^2 ).*c;
    uuhat=[uhat(m+1:N),uhat(1:m)];
    uu =ifft(uuhat);
    u(s,:,1)= [uu(m+1:N),uu(1:m)];
end


for m1=1:M;
for m2=1:N;
    V(m2)=3.*i.*(sech(x(m2)))^2*u(m1,m2,1)-i*(abs(u(m1,m2,1))).^2*(u(m1,m2,1));
end
    FF =[V(m+1:N),V(1:m)] ;
    C=fft(FF);
    S(m1,:)=[C(m+1:N),C(1:m)];
end
u(1,:,2)=zeros(1,N);
for m3=2:M
    for m4=1:m3
    Vhat = exp(-i.*(t(m3)-t(m4)).*(xi).^2 ).* S(m4,:);
    VVhat=[Vhat(m+1:N),Vhat(1:m)];
    VV =ifft(VVhat);
    V1(m4,:)=[VV(m+1:N), VV(1:m)];
    end
```

```
   u(m3,:,2)=(h/2)*(V1(1,:)+2*sum(V1(2:m3-1,:))+V1(m3,:));
end
figure(1)
[X,T] = meshgrid(x,t(11:end));
xlabel('x'); ylabel('t'); zlabel('u');
mesh(T,X,abs(u(11:end,:,1)));
U2=u(:,:,1)+u(:,:,2);
figure(2)
mesh(T,X,abs(U2(11:end,:)));
figure (3)
Uex=exp(i*t')*sech(x);
E1=max(abs(Uex'-u(:,:,1)'));
plot(t(21:end),log10(E1(21:end)));
hold on

E2=max(abs(Uex'-U2'));
plot(t(21:end),log10(E2(21:end)),'r:');
hold on

for e=2:d-1;
    for m1=1:M;
        for m2=1:N;
            W=0;
            for f=1:e;
            Z=0;
                for q=1:e-f+1
Z=Z+(conj(u(m1,m2,q))).*u(m1,m2,f).*u(m1,m2,e-f-q+2);
                end
```

```
            W=W+Z;
             end
        V(m2)=3.*i.*(sech(x(m2)))^2.*u(m1,m2,e)-i.*W;
        end
    FF =[V(m+1:N),V(1:m)] ;
    C=fft(FF);
    S(m1,:)=[C(m+1:N),C(1:m)];
end

for m3=2:M
    for m4=1:m3
    Vhat = exp(-i.*(t(m3)-t(m4)).*(xi).^2 ).* S(m4,:);
    VVhat=[Vhat(m+1:N),Vhat(1:m)];
    VV =ifft(VVhat);
    V1(m4,:)=[VV(m+1:N), VV(1:m)];
    end
   u(m3,:,e+1)=(h/2)*(V1(1,:)+2*sum(V1(2:m3-1,:))+V1(m3,:));
end
end
```

Bibliography

[1] K. Abbaoui, and Y. Cherruault, "Convergence of Adomians method applied to differential equations", Comp. Math. Appl., **28**, 103-9 (1994a).

[2] K. Abbaoui, and Y. Cherruault, "Convergence of Adomians method applied to nonlinear equations", Mathematical and Computer Modelling, **20**, 69-73, (1994b).

[3] G. Adomian, "Nonlinear stochastic systems: Theory and applications to Physics", Kluwer Academic press, (1989).

[4] G. Adomian, "Solving Frontier problem of Physics: The Decomposition Method", Kluwer Academic press, (1994).

[5] G. Adomian, Y. Cherruault, K. Abbaoui, "A nonperturbative analytical solution of immune response with time-delays and possible generalization", Math. Comput. Modelling, **24**, 89-96 (1996).

[6] K. Al-Khaled, "Numerical approximations for population growth models", Applied mathematics and computation, **160**, 865-73 (2005).

[7] K. Al-Khaled, and F. Allan, "Construction of solution for the shallow water equations by the decomposition method", Mathematics and computers in simulation, **66**, 479-86 (2005).

[8] F. Allan, and M. Syam, "On the analytic solutions of the nonhomogeneous Blasius problem", J. Comput. Appl. Math., **182**, 362-71 (2005).

[9] F. Allan, and K. Al-Khaled, "An approximation of the analytic solution of the shock wave equation", J. Comput. Appl. Math., **192**, 301-9 (2006).

[10] E. Babolian, and J. Biazar, "On the order of convergence of Adomian method", Appl. Math. Comput., **130**, 383-87 (2002).

[11] N. Bellomo, and R. A. Monaco, "Comparison between Adomians decompositions methods and perturbation techniques for nonlinear random differential equations", J. Math. Anal. Appl., **110**, 495-502 (1985).

[12] N. Bellomo, and D. Sarafyan, "On Adomians decomposition methods and some comparisons with Picard's iterative scheme", J. Math. Anal. Appl., **123**, 389-400 (1987).

[13] A. Boumenir and M. Gordon, "The Rate of Convergence for the Decomposition Method", Numerical Functional Analysis and Optimization, **25**, 15-25 (2004).

[14] Y. Cherruault, "Convergence of Adomians method", Kybernetes, **18**, 31-8 (1989).

[15] Y. Cherruault and K. Abbaoui, "A computational approach to the wave equations An application of the decomposition method", Kybernetes, **33**, 80-97 (2004).

[16] C. Chiu, and C. Chen, "A decomposition method for solving the convective longitudinal fins with variable thermal conductivity", Int. J. Heat Mass Transfer, **45**, 2067-75 (2002).

[17] J.Y. Edwards, J.A. Roberts, and N.J. Ford, "A comparison of Adomians decomposition method and RungeKutta methods for approximate solution of some predator prey model equations", Technical Rebort **309**, Mancheste Center of Computational Mathematics, 1-17 (1997).

[18] A. M. El-Sayed, and M. Gaber, "The Adomian decomposition method for solving partial differential equations of fractal order in finite domains", Physics Letters A., **359**, 175-82 (2006).

[19] S. M. El-Sayed, and M. R. Abdel-Aziz, "A comparison of Adomians decomposition method and wavelet-Galerkin method for solving integro-differential equations", Appl. Math. Comput., **136**, 151-59 (2003).

[20] I. El-Kalla, "Error Analysis of Adomian Series Solution To a Class of Nonlinear Differential Equations", Applied Mathematics E-Notes, **7**, 214-21 (2007).

[21] D. Gejji and H. Jafari, "An iterative method for solving nonlinear functional equations", J. Math. Anal. Appl., **316**, 753-63 (2006).

[22] M. A. Golberg, "A note on the decomposition method for operator equation", Appl. Math. Comput., **106**, 215-20 (1999).

[23] M. Grasselli, and D. Pelinovsky, "Numerical Mathematics", Jones and Bartlett Publishers, (2008).

[24] R. E. Greene and S. G. Krantz, "Function Theory of One Complex Variable", American Mathematical Society, **40**, (2006).

[25] S. Guellal, and Y. Cherruault, "Application of the decomposition method to identify the distrubed parameters of an elliptical equation", Math. Comput. Modelling, **21**, 51-5 (1995).

[26] S. Guellal, P. Grimalt, and Y. Cherruault, "Numerical study of Lorentz's equation by the Adomian method", Comput. Math. Appl., **33**, 25-9 (1997).

[27] N. Himoun, K. Abbaoui, and Y. Cherruault, "New results of convergence of Adomian's method", Kybernetes, **28**, 423-29 (1999).

[28] N. Himoun, K. Abbaoui, and Y. Cherruault, "Short new results on Adomian method", Kybernetes, **32**, 523-39 (2003).

[29] M. Hosseini, "Adomian decomposition method with Chebyshev polynomials ", Appl. Math. Comput., **175**, 1685-93 (2006).

[30] M. Hosseini, and H. Nasabzadeh, "On the convergence of Adomian decomposition method", Appl. Math. Comput., **182**, 536-43 (2006).

[31] M. Inc, "On numerical Solution of Partial Differential Equation by the Decomposition Method", Kragujevac J. Math., **26**, 153-64 (2004).

[32] H. Jafari, and D. Gejji, "Revised Adomian Decomposition Method for solving a system of nonlinear equations", Appl. Math. Comput., **7**, 175-181 (2006).

[33] S. Khelifa, and Y. Cherruault, "New results for the Adomian method", Kybernetes, **29**, 332-54 (2000).

[34] P. Laffez, and K. Abbaoui, "Modelling of the thermic exchanges during a drilling. Resolution with Adomian's decomposition method", Math. Comput. Modelling, **23**, 11-14, (1996).

[35] D. Lesnic, "The decomposition method for forward and backward time-dependent problems", Journal of Computational and Applied Mathematics, **147**, 27-39 (2002).

[36] X. Luo. Q. Wu. and B. Zhang, "Revisit on partial solutions in the Adomian decomposition method: solving heat and wave equations", J. Math. Anal. Appl., **321**, 353-63 (2006).

[37] X. Luo, "A two-step Adomian Decomposition Method", Appl. Math. Comput., **170**, 570-83 (2005).

[38] M. Ndour, K. Abbaoui, H. Ammar, and Y. Cherruault, "An example of an interaction model between two species", Kybernetes, **25**, 106-18 (1996).

[39] S. Pamuk, "Solution of the porous media equation by Adomians decomposition method", Phys. Lett. A., **344** 184-8 (2005).

[40] R. Rach, "On the Adomian (decomposition) method and comparisons with Picard's method", J. Math. Anal. Appl., **128**, 480-3 (1987).

[41] F. Sanchez, K. Abbaoui, and Y. Cherruault, "Beyond the thin-sheet approximation: Adomian's decomposition", Optics Commun., **173**, 397-401 (2000).

[42] N. Shawagfeh, and D. Kaya, "Comparing numerical methods for the solutions of systems of ordinary differential equations", Applied mathematics letters, **17**, 323-28 (2004).

[43] W. Tutschke, "Solution Of Initial Value Problems in Classes Of Generalized Analytic Functions", Springer-Verlag, (1989).

[44] A. Wazwaz, and S. Khuri, "The decomposition method for solving a second kind Fredholm equation with a logarithmic kernel", J. Computer Math., **61**, 103-10 (1996).

[45] A. Wazwaz, "A comparison between Adomian decomposition method and Taylor series method in the series solutions", Appl. Math. Comput., **97**, 37-44 (1998).

[46] A. Wazwaz, "A reliable modification of Adomian decomposition method", Appl. Math. Comput., **102**, 77-86 (1999a).

[47] A. Wazwaz, "Analytical approximations and Pade approximations for Volterras population model", Appl. Math. Comput., **100**, 13-25 (1999b).

[48] I. K. Youssef, "Picard iteration algorithm combined with Gauss-Seidel technique for initial value problems", Applied mathematics and Computation, **190**, 345-55 (2007).

[49] X. Zhang, "A modification of the Adomian Decomposition Method for a class of nonlinear singular boundary value problems", J Comput. Appl. Math., **180**, 377-89 (2005).

[50] Y. Zhu, Q. Chang, and S. Wu, "A new algorithm for calculating Adomian polynomials", Appl. Math. Comput., **169**, 402-16 (2005).