A Computational Approach to Physics

A Computational Approach to Physics

By

M. Ebrahim Foulaadvand

Cambridge
Scholars
Publishing

A Computational Approach to Physics

By M. Ebrahim Foulaadvand

This book first published 2023

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-5275-0126-4
ISBN (13): 978-1-5275-0126-3

Dedicated to the memory of *Jamshid Kashani* (Jamshid al-Kashi) (c1380-1429), the Persian (Iranian) mathematician and astronomer who invented the mechanical planetary computer which he called the Plate of Zones, and could graphically solve several planetary problems, including the prediction of the accurate positions in longitude of the Sun and Moon, and the planets in terms of elliptical orbits; the latitudes of the Sun, Moon, and planets; and the ecliptic of the Sun. Kashani computed the Pi number to the 16th decimal place and directed the Samarkand observatory.

Contents

Preface

The subject of *Computational Physics* has attracted much interest among physics and engineering students in recent years. It is a widely growing field thanks to the development and improvement of computational facilities namely, hard and software, clusters of computers, and parallel computation. Outstanding books on computational physics, such as the one written by Gould, Tobochnik, and Christian, from which I have learnt a lot, and the one by Landau, José Páez and Bordeianu have inevitably stimulated the students' interests to the subject. Physics is now profoundly and remarkably connected to numerical techniques and simulation methods. It requires a lot of computational skills for a proper understanding of its concepts and challenges. This has dramatically affected the traditional practice of training physics. The computational approach to physics is becoming an integral part of the physics educational programme. Many departments have included a computational physics course in their bachelor's or master's curricula. Students normally enjoy computational subjects because they realise how easier and more fascinating it will be to employ computational methods and techniques to solve complex problems in physics that are not amenable to exact analytical solutions. They mostly prefer to avoid the complexity of rigorous analytical solutions and instead enjoy solving the physical problems facilitated by packages and softwares numerically. The above opinion and viewpoint are based on the feedback I gained from students over a couple of years of experience in teaching computational physics. In many existing books on computational physics, the lack of a sufficient number of solved problems and examples is obvious. Many of these books heavily emphasise on teaching numerical techniques such as numerical differentiation, error analysis, root finding, etc., instead of applying

them to concrete physical problems. Another major shortcoming of many books with the terms "computation" and "physics" is their concentration on details of writing subroutines and codes without implementing them to sophisticated physical problems to be solved by these codes. Of course, there are exceptions, like the notable books written by Nicholas J. Giordano, and Alejandro Garcia, from which I have significantly learnt, borrowed, and benefited. On the contrary, the main goal of this book is to introduce students how to apply numerical techniques to problems of classical as well as modern physics. The book focuses and emphasizes on solving physics problems. In some sense, it can be considered as a *problem solver*. The distinguishing pedagogical feature of the book is two-fold. First, it provides many numerically solved problems and examples, and second, it briefly reviews analytical results and exact solutions as warming up before numerical involvement to problems that cannot be solved analytically. In this way, the students will achieve a better insight into the necessity of the numerical approach as a supplementary tool for the description of the underlying physics of the problem. The book level is upper intermediate and will be most appropriate for senior undergraduates, and junior postgraduate students. It is assumed that the potential reader has elementary programming knowledge. Therefore, I have avoided including the basics of computer programming and numerical methodologies such as differentiation, integration, etc. All the source codes and programme listings are in *C* programming language. You can access these codes and subroutines in the appendix, which appears online on the book's website: **http://www.znu.ac.ir/members/newpage/998/en**.

I am hopeful that the book will attract the attention of bright and interested students. Needless to say that the book is not free of typos and errors. It is highly appreciated to receive your comments and suggestions that can be sent to any of my email addresses: foolad@znu.ac.ir or ebrahim.foulaadvand@gmail.com.

Tehran, April 2023 M. Ebrahim Foulaadvand

Acknowledgement

I was attracted and inspired to Computational Physics during the "computer application in physics" undergraduate course that I passed with Reza Ejtehadi in 1998 at the former Aria Mehr (current Sharif) University of Technology in Tehran when I was a graduate student. He is gratefully appreciated. I am immensely thankful to the Institute for Research in Fundamental Sciences (IPM) at Tehran for providing me with a working office at which most of the manuscript was written. I am especially indebted to Somayyeh Belbasi, my former Ph.D. student, who contributed to several codes and subroutines. Some colleagues have provided essential advice and encouragement, including Andreas Schadschneider from the University of *Köln*, Gunter *Schütz* from the *Jülish* Forschungszentrum and Philipp Maass from the University of *Osnabrück*. Special thanks are dedicated to Ahmad Shariati and Amir Aghamohammadi from the Alzahra University of Tehran for invaluable help, advice, and technical support. Sincere thanks are given to anonymous referees who proposed valuable and constructive comments to the text. I appreciate the helps and feedback from many enthusiastic students during my lectures on Computational Physics that I gave at the University of Zanjan, Tarbiyat Modares university/Tehran, and the Institute for Advanced Studies at Basic Sciences (IASBS)/Zanjan from which this book has been evolved and emerged. It is the author's pleasure to thank the *Cambridge Scholar Publication* for the very enjoyable cooperation which made this book possible. Finally, the author deeply appreciates his wife, Azadeh, his son Aria Radin, and his daughter Arnika for their understanding, support, and patience during the preparation period of the manuscript.

Chapter 1

1st order differential equations in physics

The laws of physics are frequently formulated within the mathematical framework of differential equations. Therefore, it would be natural to begin our numerical investigations from such equations. For the sake of simplicity, let us start from the simplest case i.e.; first-order differential equations. There are a lot of topics that involve first-order equations. Examples include cooling of hot bodies, the Fourier law of heat transfer, decay of radioactive nuclei, etc. As our first topic let us numerically explore the problem of *coffee cooling* which is discussed in several computational textbooks.

1.1 First order differential equations

In this section, we try to introduce some physical problems which are mathematically formulated in terms of first-order ordinary differential equations. We begin with the old problem of coffee cooling which dates back to the time of Newton.

1.1.1 Coffee Cooling problem

If you put a cup of hot coffee on a table it will cool down until its temperature reaches the room's temperature. Experimental measurement of coffee temperature versus time proves that to a good approximation the temperature-time curve $T(t)$ obeys an exponential fall

(H. Gould and Chriastian, 2006). Energy transfer from the hot water in a cup of coffee to the surrounding air is complicated and in general involves convection, radiation, and conduction mechanisms. However, it can be shown that if the temperature difference between coffee and its surroundings is not too large, the rate of the coffee temperature change can be taken to be proportional to the temperature difference between coffee and its surrounding. Mathematically, one can formulate this statement by the following differential equation:

$$\frac{dT}{dt} = -r(T - T_s). \tag{1.1}$$

where $T(t)$ denotes the instantaneous coffee temperature and T_s denotes the surrounding (room) temperature. The constant r is the cooling constant. The minus sign in (1.1) implies that if $T > T_s$, the coffee temperature will decrease with time. The value of the cooling constant r depends on the heat transfer mechanism, the contact area with the surroundings, and the thermal properties of the water. Equation (1.1) is sometimes known as Newtons law of cooling. Although we all know the analytical solution of the linear first-order differential equation (1.1) let us re-obtain it. We show the initial coffee temperature by T_0 and perform a change of variable $T - T_s = Y$. Equation (1.1) then becomes $\frac{dY}{dt} = -rY$. In terms of the new variable Y, the initial condition becomes $Y(0) = T_0 - T_s$. We therefore, find $Y(t) = Y(0)e^{-rt}$ which in turn implies:

$$T(t) = T_0 e^{-rt} + (1 - e^{-rt})T_s \tag{1.2}$$

To solve (1.1) numerically we should be able to give $T(t)$ at discrete times t_n which are separated from each other by a fixed time interval Δt. In fact, the nth step of time will be $t_n = n\Delta t$. The next step is to write the time derivative in a finite difference form. This is simply achieved by Taylor expansion around t_n as follows:

$$T(t_n + \Delta t) = T(t_n) + \Delta t \frac{dT}{dt}(t_n) + \frac{1}{2}(\Delta t)^2 \frac{d^2T}{dt^2}(t_n) + O(\Delta t)^3 \tag{1.3}$$

Keeping only the term proportional to Δt in (1.3) one finds:

$$\frac{dT}{dt}(t_n) = \frac{T(t_n + \Delta t) - T(t_n)}{\Delta t} + O(\Delta t) \tag{1.4}$$

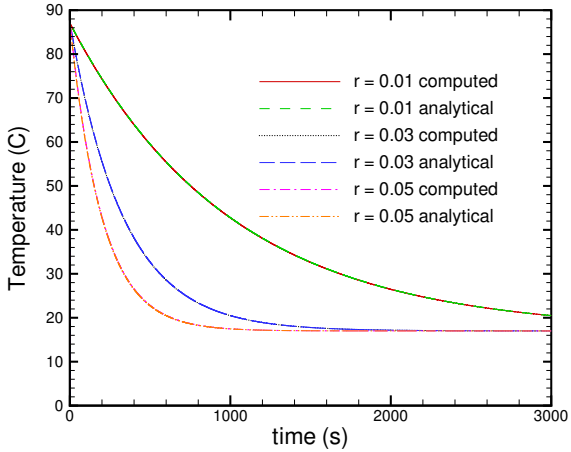

Figure 1.1: *Time evolution of coffee temperature for various values of cooling rate parameter r. Numerical results have been compared to analytical ones. The time step has been set to $\Delta = 0.1$ s. The initial and surrounding temperatures are $T_0 = 87\ ^0C$, $T_s = 17\ ^0C$.*

Putting this on the left-hand side of (1.1) and ignoring the truncation error, which is of order Δt, we simply find:

$$T_{n+1} = T_n - r\Delta t(T_n - T_s) \tag{1.5}$$

Note we have used T_n as a shorthand notation for $T(t_n)$. This is the basic equation that can be iterated from $n = 0$ to give the temperature T at subsequent steps $n = 1, 2, \cdots$. The routine `CoffeeCool` (see appendix 1.A) implements this algorithm which is known as the *Euler algorithm*. Figure (1.1) shows the numerical solution of (1.1) for some values of cooling rate r. The parameters have been set to $T_0 = 87\ ^0C$, $T_s = 17\ ^0C$, and $\Delta t = 0.1$ s. We have compared our numeric solution with the analytical one given by (1.2). You see there is a very good agreement between computed and analytical results. This shows that the Euler algorithm has a good performance in solving linear first-order differential equations.

1.1.2 Nonlinear 1st order differential equations

In the coffee cooling problem, we encountered a linear first-order differential equation which was numerically solved by the Euler algorithm. The Euler algorithm can also be implemented to solve non-linear first-order equations in the following general form:

$$\frac{dx}{dt} = f(x,t) \tag{1.6}$$

Where $f(x,t)$ is a given function. Note the independent variable t need not necessarily be time and the dependent variable x need not be position. Let x_n and f_n be shorthand notations for $x(t_n)$ and $f(x_n,t_n)$ correspondingly where t_n is the $n-th$ timestep. One can obtain the numerical solution of (1.6) according to the Euler algorithm as follows:

$$x_{n+1} = x_n + \Delta t f(x_n,t_n) \tag{1.7}$$

If we want a better approximation we should resort to higher-order terms in the Taylor expansion. To this end, we have:

$$x_{n+1} = x(t_n+\Delta t) = x(t_n)+\Delta t\frac{dx}{dt}(t_n)+\frac{1}{2}(\Delta t)^2\frac{d^2x}{dt^2}(t_n)+O(\Delta t)^3 \tag{1.8}$$

From (1.6) we can replace $\frac{dx}{dt}(t_n)$ and $\frac{d^2x}{dt^2}(t_n)$ with $f(x_n,t_n)$ and $\frac{df}{dt}(x_n,t_n)$ respectively. To evaluate the latter one we proceed as follows:

$$\frac{df}{dt}(x_n,t_n) = \frac{\partial f}{\partial x}(x_n,t_n)\frac{dx}{dt}(x_n,t_n) + \frac{\partial f}{\partial t}(x_n,t_n) \tag{1.9}$$

Using (1.6) it can be simplified as follows:

$$\frac{df}{dt}(x_n,t_n) = \frac{\partial f}{\partial x}(x_n,t_n)f(x_n,t_n) + \frac{\partial f}{\partial t}(x_n,t_n) \tag{1.10}$$

Therefore, the second-order Taylor expansion gives:

$$x_{n+1} = x_n + \Delta t f(x_n,t_n) + \frac{1}{2}(\Delta t)^2[\frac{\partial f}{\partial x}(x_n,t_n)f(x_n,t_n)+$$

$$\frac{\partial f}{\partial t}(x_n,t_n)] + O(\Delta t)^3 \tag{1.11}$$

Given f all the terms on the right-hand side of (1.11) can be evaluated. We can go to higher terms in the Taylor expansion but the manipulations become more cumbersome. For more details, see other references (Pang, 2006; Scherer, 2010; Rubin H. Landau and Bordeianu, 2008).

1.1.3 Radioactive decay

Another example that involves a first-order linear differential equation is the decay of radioactive nuclei. Many unstable nuclei can decay into smaller atoms. This phenomenon is random in nature and we can only speak of the decay probability. A typical example is the nucleus of a Uranium isotope ^{238}U which can decay into a lighter atom ^{234}Th. Suppose we have $N(t)$ Uranium atoms at time t. If we denote the decay rate by λ, during the infinitesimal time interval $[t, t + \Delta t]$ on average a small number of Uranium atoms decay. This number will be proportional to the number of atoms $N(t)$ and the time interval Δt. The proportionality constant is the decay rate λ. Therefore we can write the following equation:

$$N(t) - N(t + \Delta t) = \lambda \Delta t N(t) \tag{1.12}$$

Even though the number of nuclei should be an integer number but we often can treat this number as a continuous variable and end up with the following first-order linear differential equation in the limit $\Delta t \to 0$:

$$\frac{N(t + \Delta t) - N(t)}{\Delta t} = \frac{dN(t)}{dt} = -\lambda N(t) \tag{1.13}$$

Equation (1.13) can simply and analytically be solved. The solution turns out to be:

$$N(t) = N_0 e^{-\lambda t} \tag{1.14}$$

In which N_0 is the initial number of uranium Nuclei at $t = 0$. Note the decay rate λ has the dimension of inverse time. We can solve (1.13) numerically as we did for the coffee cooling problem. Denoting $N(t_n)$ by N_n we can turn (1.13) into a finite difference equation:

$$\frac{N_{n+1} - N_n}{\Delta t} = -\lambda N_n \to N_{n+1} = N_n(1 - \lambda \Delta t) \tag{1.15}$$

The routine `NuclearDecay` (see appendix 1.B for details) implements the above Euler algorithm to find the numerical solution of (1.13). Figure (1.2) compares the numerical solutions with the analytical ones for some choices of decay rate λ. As you see there is an excellent agreement between numerical and analytical solutions. A useful concept in nuclear decay is half-life $T_{\frac{1}{2}}$ which is the time that half of the nuclei

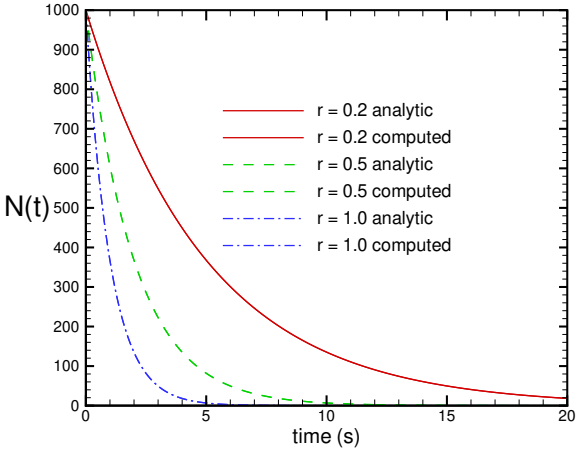

Figure 1.2: *Computed number of undecayed nuclei versus time for various values of the decay constant λ by Euler algorithm with $\Delta t = 0.01$. The number of radioactive nuclei has been initially set to $N_0 = 1000$. Comparison is made with the analytical solution.*

decay. Theoretically, it is related to decay rate λ as $T_{\frac{1}{2}} = \frac{ln2}{\lambda}$. Numerically we can compute $T_{\frac{1}{2}}$ via a simple `if` command in the subroutine. In the time loop, we can simply find the timestep t^* at which the condition $N(t^*) = \frac{N_0}{2}$ is satisfied. Half-life is then found to be $T_{\frac{1}{2}} = t^* \Delta t$. A more realistic situation is the double decay process. Let us pose this problem exactly as it is presented in problem four from chapter one of the book (Giordano and Nakanishi, 2006).

Problem:

Consider a radioactive decay problem involving two types of nuclei A and B with population $N_A(t)$ and $N_B(t)$. Suppose type A nuclei decay to type B which then also decay according to the following differential equations:

$$\frac{dN_A(t)}{dt} = -\lambda_A N_A(t) \tag{1.16}$$

$$\frac{dN_B(t)}{dt} = \lambda_A N_A(t) - \lambda_B N_B(t) \tag{1.17}$$

where λ_A and λ_B are decay constants for each type of nucleus. Use the Euler method to solve these coupled equations for $N_A(t)$ and $N_B(t)$ as a function of time. Explore the behaviour found for different values of $\frac{\lambda_A}{\lambda_B}$.

Before trying to solve the problem numerically the exact analytical solution is presented. Equations (1.16) and (1.17) form a linear set of first-order differential equations. Assume at $t = 0$ there are N_{0A} and N_{0B} nuclei respectively. Equation (1.16) simply gives:

$$N_A(t) = N_{0A} e^{-\lambda_A t} \tag{1.18}$$

Replacing $N_A(t)$ from (1.18) into (1.17) we find:

$$\frac{dN_B(t)}{dt} = \lambda_A N_{0A} e^{-\lambda_A t} - \lambda_B N_B(t) \tag{1.19}$$

Equation (1.19) is an inhomogeneous first-order differential equation for $N_B(t)$. Its solution comprises the sum of a special solution of the inhomogeneous equation plus an arbitrary solution of the homogeneous equation. To find the special solution we assume $N_B^s(t) = ae^{bt}$. Putting this into (1.19) we arrive at $abe^{bt} = \lambda_A N_{0A} e^{-\lambda_A t} - \lambda_B a e^{bt}$. The only way for this equation to be satisfied at an arbitrary time t is that $b = -\lambda_A$ to have a common exponential factor $e^{-\lambda_A t}$ on both sides. With this b the unknown a turns out to be $a = \frac{\lambda_A}{\lambda_B - \lambda_A} N_{0A}$ which then gives the special solution as follows: $N_B^s(t) = \frac{\lambda_A}{\lambda_B - \lambda_A} N_{0A} e^{-\lambda_A t}$. Having found the special solution, $N_B(t)$ can be written as follows:

$$N_B(t) = N_B^s(t) + Ce^{-\lambda_B t} \tag{1.20}$$

where the constant C is determined by requiring that $N_B(t)$ satisfies the initial condition $N_B(0) = N_{0B}$. Constant C simply is found to be: $C = N_{0B} - \frac{\lambda_A}{\lambda_B - \lambda_A} N_{0A}$. Putting this C into (1.20) $N_B(t)$ becomes:

$$N_B(t) = \frac{\lambda_A}{\lambda_B - \lambda_A} N_{0A} (e^{-\lambda_A t} - e^{-\lambda_B t}) + N_{0B} e^{-\lambda_B t} \tag{1.21}$$

To solve the problem numerically we show the number of nuclei A and B in timestep n by N_A^n and N_B^n respectively. The application of the

Euler algorithm with a timestep τ gives:

$$N_A^{n+1} = N_A^n - \tau\lambda_A N_A^n = N_A^n(1-\tau\lambda_A) \tag{1.22}$$

$$N_B^{n+1} = N_B^n + \tau(\lambda_A N_A^n - \lambda_B N_B^n) = N_B^n(1-\tau\lambda_B) + \tau\lambda_A N_A^n \tag{1.23}$$

Figure (1.3) sketches the computed N_A and N_B versus time obtained by the programme `DoubleNuclearDecay` (see appendix 1.C for details). Comparison to the analytical solution is also shown. The parameters are chosen as follows: $N_{0A} = 1000, N_{0B} = 200, \lambda_A = 0.2$ and $\lambda_B = 0.1$. As you see N_B increases first and then after reaching a maximum and then it decreases exponentially to zero. The reason is that we have chosen $\lambda_A > \lambda_B$ so that in the early stages of the decay process, the number of decayed nuclei of type A exceeds the number of B nuclei therefore $N_B(t)$ increases. After a sufficient time, there remain fewer A nuclei hence the source of B nuclei production smears off, and the decay process dominates the production. There is an excellent agreement between computed and analytical solutions. We see the Euler algorithm also shows a successful performance in dealing with a linear set of first-order differential equations.

1.2 Nonlinear differential equation: Population Growth

As another application of first-order differential equations, we consider the growth and extinction of populations. In contrast to previous examples, the growth problem often involves nonlinear differential equations. Let us begin with a simple solvable nonlinear problem which is posed as a problem in chapter one of (Giordano and Nakanishi, 2006). Suppose the number of individuals in a population is $N(t)$ at time t. The population number can grow over time through the birth process. The number of newly born individuals per time unit is assumed to be proportional to $N(t)$ with proportionality constant a. The population number can decrease over time due to the death process. The death rate can be proportional to $N^2(t)$ to allow for the fact that food will become harder to find when the population number becomes larger. One can then write the following nonlinear rate equation:

$$\frac{dN(t)}{dt} = aN(t) - bN^2(t) \tag{1.24}$$

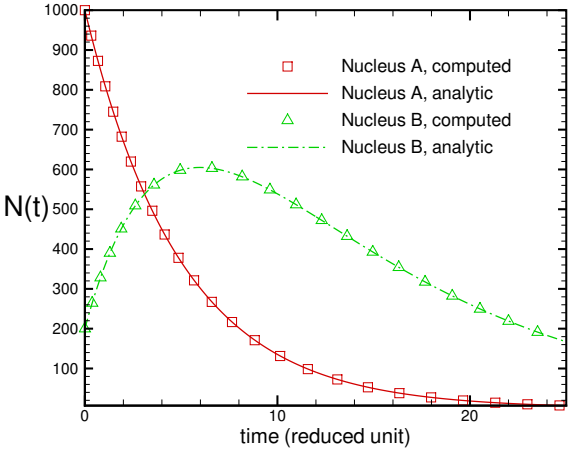

Figure 1.3: *Number of undecayed nuclei A and B versus time. The Euler algorithm with a timestep $\tau = 0.01$ has been implemented. Comparison is made with analytical solution. Decay rates are $\lambda_A = 0.2$ and $\lambda_B = 0.1$.*

Hopefully, equation (1.24) can be solved exactly. For this purpose, we simply write it as $\frac{dN}{aN-bN^2} = dt$ and integrate it from both sides. Supposing that at $t = 0$ the population number is $N(0) = N_0$. We have:

$$\frac{dN}{aN - bN^2} = dt \Rightarrow \int_{N_0}^{N(t)} \frac{dN}{aN - bN^2} = \int_0^t dt = t \tag{1.25}$$

Concerning the identity $\frac{1}{aN-bN^2} = \frac{1}{aN} + \frac{b}{a^2-abN}$ we find:

$$t = \int_{N_0}^{N(t)} (\frac{dN}{aN} + \frac{bdN}{a^2 - abN}) = \frac{1}{a}[ln\frac{N(t)}{N_0} - ln\frac{a - bN(t)}{a - bN_0}] \tag{1.26}$$

Equation (1.26) gives: $at = ln\frac{(a-bN_0)N(t)}{N_0[a-bN(t)]}$ which then implies:

$$\frac{N_0}{a - bN_0}e^{at} = \frac{N(t)}{a - bN(t)} \tag{1.27}$$

A simple algebra gives $N(t)$ as follows:

$$N(t) = \frac{aN_0 e^{at}}{a + bN_0(e^{at} - 1)} \tag{1.28}$$

In the long-time limit $t \gg \frac{1}{a}$ one finds $N \to N_s = \frac{a}{b}$. Note that we could obtain the stationary solution by setting the left-hand side of (1.24) equal to zero. Now let us see to what extent the numerical solution agrees with the exact one. As usual, we incorporate the Euler algorithm which is implemented as $\frac{N^{n+1}-N^n}{\tau} = aN^n - b(N^n)^2$. We find:

$$N^{n+1} = N^n(1 + a\tau - b\tau N^n) \tag{1.29}$$

Figure (1.4) shows the numerical solution, based on the iteration in (1.29). It has been obtained by the programme `PopulationGrowth` (see Appendix 1.D for more details). The numerical solution is also compared to the analytical solution. As you see the agreement between computed and analytical results is still very good. We have chosen the time unit such that $a = 1$ and have varied the death rate b. The initial population number is set to $N_0 = 1000$. Let us see what happens if N_0 becomes smaller. Figure (1.5) shows $N(t)$ for $N_0 = 5$. Here you see a different behaviour. Since the initial population is small the death term i.e.; $-bN^2$ would be very small in the early stages of the dynamics and therefore the birth term gives rise to the population increase. After a sufficient time, when the population has grown considerably, the death rate becomes large and does not allow for further population increase. As a result of competition between birth and death processes, the system comes to a stationary behaviour at $N_s = \frac{a}{b}$.

1.3 Master equation

Another common application of first-order differential equations appears in random processes. Suppose the outcome of an event can be realised in M different ways. The best example is throwing a die. In this case, the outcome event is the number shown by the top face of the dice. The number can be one to six ($M = 6$). When the stochastic dynamics occur in continuous time one can speak from the rate which is the probability of occurrence of an event per time unit. Let $P(n, t)$ denote the probability that at time t the nth outcome occurs

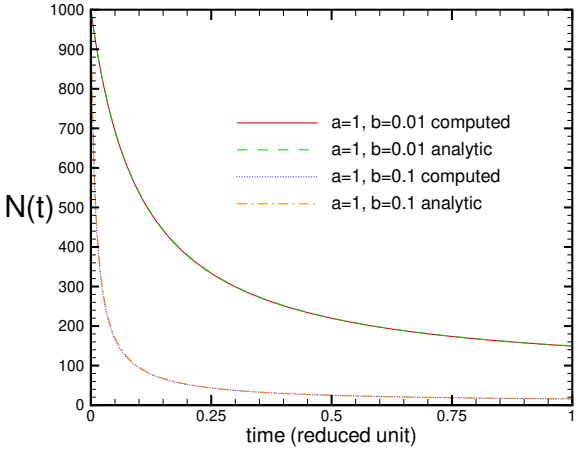

Figure 1.4: *Number of population individuals versus time with* $N_0 = 1000$. *The Euler algorithm with a timestep* $\tau = 0.001$ *has been used. Comparison is made with analytical solution.*

$(n = 1, 2, \cdots, M)$. Let $w_{n,m}$ denote the probability of transition from state n to state m $(m \neq n)$ during the infinitesimal interval $[t, t + \delta t]$. One can write the following first-order differential equation for $P(n, t)$ (van Kampen, 1992):

$$\frac{\partial P(n,t)}{\partial t} = \sum_{m \neq n} [P(m,t)w_{m,n} - P(n,t)w_{n,m}] \tag{1.30}$$

To see the derivation of (1.30) you may refer to any standard textbook on stochastic dynamics and statistical physics. I can suggest a very good one: (Reichel, 1998) (see chapter five). Equation (1.30), known as the *Master equation*, is linear and first order but the subtle point is that it is not local in the discrete state variable n and that is the point that makes its solution hard to find, if not impossible, in general. In a few special cases, we can analytically solve the Master equation. Despite our goal is not to solve the problems analytically but for didactic purposes, we prefer to start with the problems which are amenable to exact solutions. As one of these examples let us come

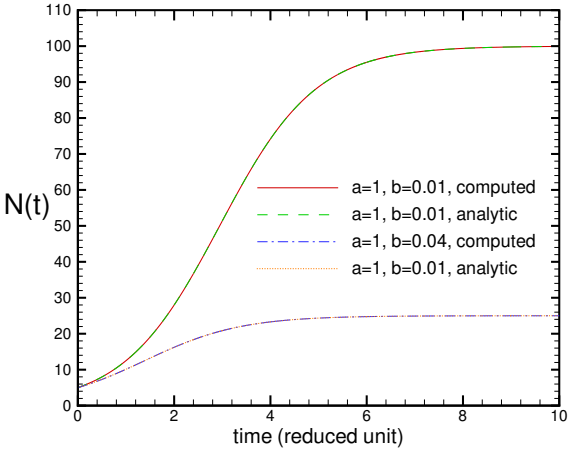

Figure 1.5: *Number of population individuals versus time with a smaller initial population number $N_0 = 5$. The Euler algorithm with a timestep $\tau = 0.001$ has been implemented.*

back again to the birth-death process. Suppose we have a population. In a more realistic description, the population number at time t can be any number $0 \leq n < \infty$. During the infinitesimal time interval δt the population number n can change by one number. It can increase to $n+1$ if a birth process occurs or decrease to $n-1$ if a death process happens. For simplicity, we assume that the birth and death rates depend linearly on the current population number n. The proportionality constants are taken as β and γ respectively. In terms of the transition rates, the only non-zero ones are those in which n and m differ by one. Therefore we have $w_{n,n+1} = \beta n$ and $w_{n,n-1} = \gamma n$. We can now write the Master equation for this random process:

$$\frac{\partial P(n,t)}{\partial t} = \beta(n-1)P(n-1,t)+\gamma(n+1)P(n+1,t)-(\beta+\gamma)nP(n,t) \tag{1.31}$$

We recall that equation (1.31) is not local in variable n and this makes the analytic solution cumbersome. However, it is possible to exactly find the mean number of the population and its deviation from the

mean value. This is elegantly done by the method of generating function in chapter five of (Reichel, 1998). We refer interested readers to this excellent book for details and only quote the main results here. The mean number of population $\langle n(t)\rangle$ is defined as follows:

$$\langle n(t)\rangle = \sum_{n=-\infty}^{\infty} nP(n,t) \tag{1.32}$$

It should be noted that the physical range for n starts from zero and we should simply set $P(n,t) = 0$ for $n < 0$. Defining a generating function $F(z,t)$:

$$F(z,t) = \sum_{n=-\infty}^{\infty} z^n P(n,t) \tag{1.33}$$

We simply find:

$$\langle n(t)\rangle = \frac{\partial F(1,t)}{\partial z}; \quad \langle n^2(t)\rangle - \langle n(t)\rangle = \frac{\partial^2 F(1,t)}{\partial z^2} \tag{1.34}$$

Higher moments are obtained similarly. With the help of the Master equation (1.31) and the definition of generating function $F(z,t)$ we can obtain the following first-order partial differential equation for $F(z,t)$ (see (Reichel, 1998) for details).

$$\frac{\partial F(z,t)}{\partial t} = (z-t)(\beta z - \gamma)\frac{\partial F(z,t)}{\partial z} \tag{1.35}$$

It is possible to solve (1.35) by the method characteristics (Sneddon, 1957). This method is explained in many mathematical physics books such as (Myint-U and Debnath, 2007). Another good reference is (Hassani, 2013). The application of this method to (1.35) with initial condition $P(n,0) = \delta_{m,n}$ gives:

$$F(z,t) = \left(\frac{\gamma(z-1)e^{(\beta-\gamma)t} - \beta z + \gamma}{\beta(z-1)e^{(\beta-\gamma)t} - \beta z + \gamma}\right)^m \tag{1.36}$$

Taking the partial derivatives of $F(z,t)$ with respect to variable z according to (1.34) gives:

$$\langle n(t)\rangle = me^{(\beta-\gamma)t} \tag{1.37}$$

$$\langle n^2(t)\rangle - \langle n(t)\rangle^2 = m(\frac{\gamma+\beta}{\gamma-\beta})e^{(\beta-\gamma)t}(1-e^{(\beta-\gamma)t}) \tag{1.38}$$

Now it is about time we solved the problem numerically and enjoy its beauty and convenience. Let us see if our Euler algorithm can give a satisfactory result or not. Because we have used n to denote the population number we denote the time step counter by k. Let P_n^k be a short notation for $P(n, k\tau)$ where τ is the timestep. The Master equation (1.31) turns into the following finite difference form:

$$P_n^{k+1} = P_n^k + \tau\beta(n-1)P_{n-1}^k + \tau\gamma(n+1)P_{n+1}^k - \tau(\beta+\gamma)nP_n^k \tag{1.39}$$

The discretised initial condition implies $P_n^0 = \delta_{m,n}$. Equation (1.39) allows for finding the probabilities at time step $k+1$ from their values at time step k. Nevertheless, to implement the Euler scheme to (1.39) we encounter a problem immediately! To evaluate P_0^1 we need to know the unphysical quantity P_{-1}^0. We need not worry because we know the physical range of n does not include negative numbers so we can remedy this problem by introducing a boundary condition $P(n,t) = 0$ whenever n becomes negative. But be careful! there is another problem in front of us. What should be done with the upper limit of n? In principle, n can go to infinity but we cannot treat this infinite limit by computer and will have to consider a finite upper limit for n. Let us suppose $n \leq M$ where M should be taken as large as possible. Now another problem arises when we want to evaluate P_M^1. In fact (1.39) gives the unphysical term P_{M+1}^0. A plausible assumption could be to set $P_{M+1}^0 = P_M^0$. More generally by setting:

$$P_{-1}^k = 0; \quad P_{M+1}^k = P_M^k \tag{1.40}$$

We can iterate (1.39) to any desired time step. The right-hand side of equation (1.40) can approximately be a finite difference form of the Neumann boundary condition $\frac{\partial P(\infty,t)}{\partial n} = 0$. Another possibility (method two) is to set $P_{M+1}^k = 0$. Specifying the right boundary condition we can iterate (1.39) in time and find $P_n^k \quad k = 1, 2, \cdots$ in the physical range $n = 0, 1, 2, \cdots$. The programme `PopulationMaster` (see appendix 1.E for details) implements the Euler algorithm with a time step $\tau = 0.01$ for solving the Master equation (1.31) numerically. Figure (1.6) exhibits the numerical solution for case $\beta = 0.1$ and $\gamma = 0.2$. We have set $M = 100$ and $m = 20$ (initial number of

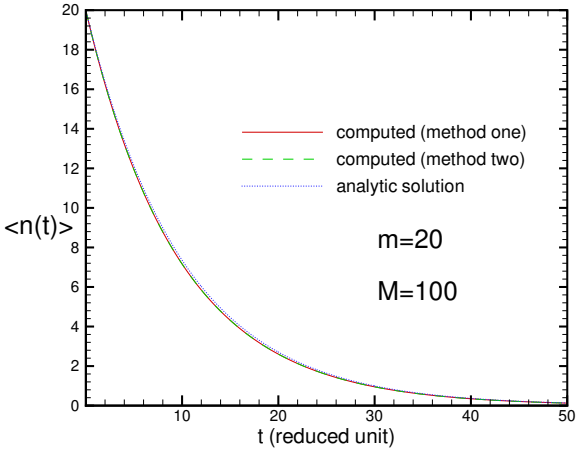

Figure 1.6: *The average number of population* $\langle n(t)\rangle$ *versus time. The Euler algorithm with a timestep* $\tau = 0.01$ *is used. The parameters are* $\beta = 0.1, \gamma = 0.2, M = 100$ *and* $m = 20$.

population). You see both methods one and two give a result that is in excellent agreement with the analytical solution (1.37). Now let us interchange the values of β and γ that is to set $\beta = 0.2$ and $\gamma = 0.1$. Figure (1.7) shows the result. We see entirely different behaviour. Now the computed answer deviates notably from the analytical one. The reason is certainly not due to the weakness of the Euler algorithm but to the inappropriateness of the right boundary condition. In the case where the birth rate is larger than the death rate, ($\beta > \gamma$) we know from the analytic solution that the average number of the population grows exponentially in time (see equation (1.37)). Accordingly, we can conclude that when t has increased the probability to have a larger population increases. This implies that for a fixed time t, the larger the n the larger $P(n,t)$ will be. Consequently, the boundary condition for P^k_{M+1} should be treated carefully. Can you suggest a more reasonable condition to reproduce the large-time behaviour of the exponentially growing population? As our final example of the birth-death process, we consider the following non-linear problem (Nicolis and Prigogine, 1977). This time let us formulate the problem in the context of chem-

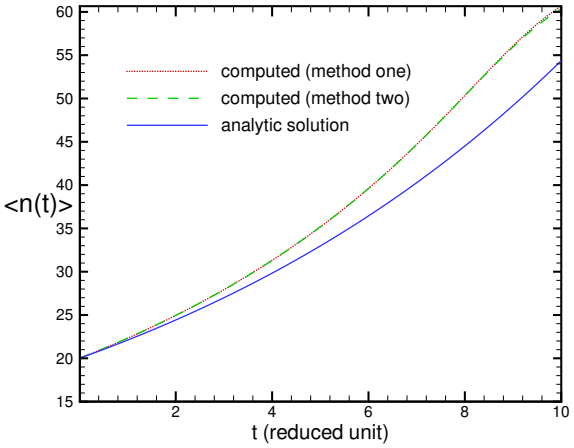

Figure 1.7: *The average number of population* $\langle n(t)\rangle$ *versus time. The Euler algorithm with a timestep* $\tau = 0.01$ *is used. The parameters are* $\beta = 0.2, \gamma = 0.1, M = 100$ *and* $m = 20$.

ical reactions. Consider a binary chemical reaction that occurs with the rate k (D. A. Mcquarrie and Russel, 1994).

$$2X \to B \tag{1.41}$$

The transition rate for this reaction is proportional both to the rate k and the number of different ways $\frac{1}{2}n_X(n_X-1)$ a pair of X molecules react to form a B molecule if there are n_X molecules in the system. Suppose before the reaction the number of X and $B-$type molecules are n_X+2 and n_B-1 respectively. After the reaction, we have n_X and n_B molecules of types X and B respectively. The transition rate turns out to be $\frac{k}{2}(n_X+2)(n_X+1)$. We can write the following Master equation for the probability to have n molecules of type X at time t:

$$\frac{\partial P(n,t)}{\partial t} = \frac{k}{2}(n+2)(n+1)P(n+2,t) - \frac{k}{2}n(n-1)P(n,t) \tag{1.42}$$

The natural boundary condition implies $P(-1,t) = 0$. Let us solve the problem numerically by the Euler method. Figure (1.8) shows the