A COMPACT FINITE DIFFERENCE METHOD OF LINES FOR SOLVING NON-LINEAR PARTIAL DIFFERENTIAL EQUATIONS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mathematics

by
Shodijon ISMOILOV

September 2022
İZMİR

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Dr. Gamze TANOĞLU for her moral and emotional support. I also could not have undertaken this journey without my co-advisor Prof. Dr. Gürhan GÜRARSLAN, who generously provided knowledge and expertise. I am extremely thankful to them for their guidance and patience, this endeavour would not be possible without them.

I sincerely thank the defense committee. Additionally, I feel honored to work with all the professors and research assistants during my graduate study.

I would like to extend my sincere thanks to all my friends especially to Sabahat Defne PLATTÜRK, Leyla AKBALIK and Mehmet KAYMAK for their friendly emotional support. I feel extremely lucky to have such friends who are encouraging, inspiring and caring.

Lastly, I am deeply indebted to my dear family for their generous financial support and patience during my education. Their belief in me has kept my spirit and motivation high during this process.

ABSTRACT

A COMPACT FINITE DIFFERENCE METHOD OF LINES FOR SOLVING NON-LINEAR PARTIAL DIFFERENTIAL EQUATIONS

In this thesis, an efficient numerical method is proposed for the numerical solution of the chemical reaction-diffusion model governed by a non-linear system of partial differential equations known as a Brusselator model. The method proposed is based on a combination of higher-order Compact Finite Difference schemes and stable time integrator known as an adaptive step-size Runge-Kutta method. The performance of adaptive step-size Runge-Kutta formula of fifth-order accurate in time and Compact Finite Difference scheme of sixth-order in space are investigated. The method is implemented to solve three test problems and reveals that the method is capable of achieving high efficiency, accuracy and reliability. The results obtained are sufficiently accurate compared to some available results in the literature.

ÖZET

DOĞRUSAL OLMAYAN KISMİ DİFERANSİYEL DENKLEMLERİN ÇÖZÜMÜ İÇİN BİR KOMPAKT SONLU FARKLAR ÇİZGİLER YÖNTEMİ

Bu tezde, kimyasal reaksiyon-difüzyon modeli olan ve Brusselator Modeli olarak da bilinen doğrusal olmayan kısmi diferansiyel denklem sisteminin çözümü için bir sayısal yöntem önerilmiştir. Önerilen yöntem yüksek mertebeden Kompakt Sonlu Fark şemaları ve kararlı zaman tümlev alıcısı, bilinen adıyla adaptif hesap adımlı Runge Kutta yönteminin kombinasyonuna dayanmaktadır. Uzayda altıncı mertebeden Kompakt Sonlu Fark Şeması ve zamanda ise beşinci mertebeden Adaptif Hesap Adımlı Runge-Kutta yönteminın performansı incelenmiştir. Yöntem üç adet test problemine uygulanmıştır ve yüksek hassasiyette(doğrulukta) çözümler elde edilmiştir. Aynı zamanda yöntemin verimli ve güvenilir olduğu ortaya çıkmıştır. Elde edilen sonuçlar literatürdeki diğer sonuçlarla uyuşmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES vii

LIST OF TABLES viii

CHAPTER 1. INTRODUCTION 1

1.1. Introduction 1

1.2. Outline of Thesis 3

CHAPTER 2. COMPACT FINITE DIFFERENCE METHOD 5

2.1. The Construction of Compact Finite Difference Schemes for the First-Order Derivatives 5

2.2. The Construction of Compact Finite Difference Schemes for the Second-Order Derivatives 12

2.3. One-Sided Approximation 15

2.3.1. Boundary Formulation for the First-Order Derivative 15

2.3.2. Boundary Formulation for the Second-Order Derivative 18

CHAPTER 3. ADAPTIVE STEP-SIZE RUNGE-KUTTA METHOD 23

3.1. A Fifth-order Adaptive Step-size Runge-Kutta Method 23

CHAPTER 4. STABILITY ANALYSIS 27

CHAPTER 5. NUMERICAL SIMULATIONS 32

CHAPTER 6. SUMMARY AND CONCLUSION 44

REFERENCES 45

APPENDIX A. MATLAB CODES FOR NUMERICAL RESULTS 49

A.1. Codes for Problem 5.1 49

A.2. Codes for Problem 5.2 52

A.3. Codes for Problem 5.3 55

A.4. Approximation Matrix for Second-order Derivative 59

LIST OF FIGURES

Figure **Page**

Figure 4.1. For $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.0001$ 4.1a gives the eigenvalues where $N = 30$ and 4.1b gives the maximum eigenvalue as N increases. Similarly for $\hat{\alpha} = 0.5$, $\hat{\beta} = 1.2$ and $\hat{\gamma} = 0.0001$ in 4.1c and 4.1d,respectively. 31

Figure 5.1. Physical behaviour of two species u and v at different times for Problem 5.1 for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.0001$. 32

Figure 5.2. The solution profile of u and v of Problem 5.1 fixed at $(x, y) = (0.5, 0.5)$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$. 34

Figure 5.3. The path taken by (u, v) as time increase (left) and the (u, v)-plane (right) of Problem 5.1 fixed at $x = 0.5$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.0001$. 34

Figure 5.4. The solution profile of u and v of Problem 5.1 fixed at $x = 0.5$ over $0 < t \leq 15$ for $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.0001$. 35

Figure 5.5. The solution profile of u and v of Problem 5.2 fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 15$ for $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.002$. 36

Figure 5.6. The physical behaviour of concentration of v in 3D and its contour of Problem 5.2 at times $t = 5, 7, 8, 9, 10.5, 12.2, 14.2, 15$. 37

Figure 5.6. continued 38

Figure 5.7. The solution profile of u and v of Problem 5.2 fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$. 39

Figure 5.8. The path taken by (u, v) as time increase (left) and the (u, v)-plane (right) of Problem 5.2 fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$. 39

Figure 5.9. The physical behaviour of concentration of u and v in 3D and contour of Problem 5.3 at time $t = 2$. 43

Figure 5.10. The solution profile of u and v of Problem 5.3 fixed at $(x, y) = (0.4, 0.6)$ over $0 < t \leq 10$. 43

LIST OF TABLES

Table **Page**

Table 2.1. Selected compact finite difference schemes for the first-order derivative. 14

Table 2.2. Selected compact finite difference schemes for the second-order derivative. 14

Table 2.3. Selected one-sided approximations for the boundary node $i = 1$ for the first-order derivative. 17

Table 2.4. Selected one-sided approximations for the boundary node $i = N$ for the first-order derivative. 17

Table 2.5. Selected one-sided approximations for the boundary node $i = 1$ and $i = N$ for the second-order derivative. 22

Table 3.1. Butcher array for explicit embedded RK formulas. 26

Table 5.1. Concentration values for oscillatory behaviour of Problem 5.1. 33

Table 5.2. The values of u and v converges to $\hat{\alpha}$ and $\hat{\beta}/\hat{\alpha}$, resp., as time increases. . 36

Table 5.3. L_∞ and relative error norms at $t = 2$ with the accuracy tolerance $\epsilon = 10^{-4}$. 40

Table 5.4. L_∞ norm errors of Problem 5.3 at $t = 2$ with the accuracy tolerance $\epsilon = 10^{-4}$. 41

Table 5.5. Comparison of L_∞ norm errors of Problem 5.3 at $t = 2$ with different accuracy tolerances. 41

Table 5.6. Comparison of numerical results produced by presented method(PM) with the results of (Ali et al., 2010) at the point $(0.40, 0.60)$. 42

Table 5.7. The values of u and v with the average of Δt's (Avg(Δt)) and CPU times at $t = 2$ with different accuracy tolerances. 42

CHAPTER 1

INTRODUCTION

In nature, many phenomena exhibit substantially non-linear behavior. These phenomena are modelled by non-linear partial differential equations (NPDE) appearing in engineering and science. Since finding the exact solution of NPDEs is a great challenge, developing numerical methods to approach the solution of these models is of high importance. The aim of this thesis is to propose an approach for numerical solutions of nonlinear reaction-diffusion problems .

1.1. Introduction

In characterization of biological and chemical reactions a process which plays a significant role is diffusion. Mathematical modelings involving this process are of high importance to study a wide range of patterns of chemical species (Mittal et al., 2022). One of the reaction-diffusion systems describing such patterns is so-called Brusselator system. Initially, the Brusselator model was proposed by Prigogine and R. Lefever in 1968 (Prigogine and Lefever, 1968; Lefever and Nicolis, 1971; Prigogine and Nicolis, 1985). The system consists of two variables interrelated with reactant and product chemicals whose concentrations are controlled (Ali et al., 2010). Certain processes that these equations model can be seen in plasma and laser physics in multiple coupling between modes, in enzymatic reactions, in formation of turing pattern on animal skin and in formation of ozone by atomic oxygen through a triple collision (Kumar et al., 2019; Mittal and Jiwari, 2011). Since, the analytical solutions of these equations are not found yet, they are of interest from the numerical point of view.

The general reaction-diffusion Brusselator system is the non-linear system of partial differential equations

$$\begin{aligned}\frac{\partial u}{\partial t} &= \hat{\alpha} - (\hat{\beta}+1)u + u^2 v + \hat{\gamma}\nabla^2 u(x,t) \\ \frac{\partial v}{\partial t} &= \hat{\beta}u - u^2 v + \hat{\gamma}\nabla^2 v(x,t)\end{aligned} \tag{1.1}$$

with initial conditions and Neumann boundary conditions

$$u(x,0) = f(x), \quad v(x,0) = g(x), \quad x \in \mathfrak{D} \tag{1.2}$$

$$\frac{\partial u(x,t)}{\partial n} = \frac{\partial v(x,t)}{\partial n} = 0, \quad (x,t) \in \partial\mathfrak{D} \times [0,T]. \tag{1.3}$$

where u and v represent concentration of two reactants, $\hat{\alpha}$ and $\hat{\beta}$ are constants of concentration of two reactants, $\hat{\gamma}$ is the diffusion coefficient and ∇^2 is Laplace operator. The studies have shown that for small values of diffusion coefficient $\hat{\gamma}$, the steady state solution of the model converges to its equilibrium points $(\hat{\alpha}, \hat{\beta}/\hat{\alpha})$ if $\hat{\alpha}^2 - \hat{\beta} + 1 > 0$.

The model has been studied and analyzed by a lot of researchers throughout the years. Various types of approaches for one-dimensional and two-dimensional case of the system are proposed to get numerical simulations of the model. A second-order method in (Twizell et al., 1999) and the decomposition method in (Wazwaz, 2000) are applied to the model whereas Ang uses the dual-reciprocity boundary element method in (Ang, 2003) to approach the solution of the system. Different types of meshfree algorithms have been developed, the approach that Al-Islam and his group (Ali et al., 2010) have taken is by combining radial basis multiquadric functions(RBMFs) and first-order finite difference method while the meshfree algorithm developed by Kumar and his coworkers (Kumar et al., 2019) is based on RBMFs combined with differential quadrature technique to get numerical solution of this model. Methods based on B-spline functions have been also widely used throughout the years. A modified trigonometric cubic B-spline functions coupled with differential quadrature method has been applied to this model by Alqahtani in (Alqahtani, 2018). Two different modified cubic B-spline based on differential quadrature algorithm have been studied in (Jiwari and Yuan, 2014; Mittal and Rohila, 2016).

Onarcan and his team (Onarcan et al., 2018) also have developed a cubic trigonometric B-spline interrelated with Crank-Nicholson method in time. A numerical technique based on Lucas and Fibonacci polynomials coupled with finite difference method and polynomial based differential quadrature method are studied to approach the solution of the model in (Haq et al., 2021; Mittal and Jiwari, 2011), respectively. Some of the popular approaches are based on fractal-fractional differential operators. In (Saad, 2021) the classical differential operators has been replaced by fractal-fractional differential operators due to the power law, exponential decay, and the generalized Mittag-Leffler kernels and the results are fairly accurate. Jena and his colleagues (Jena et al., 2020) have developed a semi-analytical technique called fractional reduced differential transform method characterized by the time-fractional derivative to approximate the solution of Brusselator model.

The main purpose of this thesis is to propose an alternative approach for solving reaction-diffusion Brusselator model. The approach that we and most authors undertake is so called the method of lines. The method of lines is a technique to reduce a PDE into an ODE by discretizing all the differentials but one. In our case, the Compact Finite Difference method is used to discretize the spatial derivatives of the model and leave the time variable continuous. Now that the model is converted into a time-dependent ODE, an adjustment step-size algorithm called an adaptive step-size Runge-Kutta method based on embedded Runge-Kutta formulas is applied to integrate the time.

1.2. Outline of Thesis

The organization of the outline of this thesis is as follows;

Chapter 2 explains the formulation of Compact Finite Difference Method. While some special schemes are formulated to provide a background about the implementation of the method, a wide range of options for derivation of different order schemes is presented. A compact scheme of sixth order is constructed to approximate the derivative in space.

Chapter 3 deals with an ODE time integrator. The type of time integrator that we use is known as an adaptive step-size Runge-Kutta method based on embedded Runge-Kutta formulae. The construction of the step-size controller is shown according to the

Butcher array which gives the coefficients for a fifth-order adaptive step-size Runge-Kutta.

Chapter 4 analyzes the stability of our method implemented to the Brusselator model. After spatial discretization the system (1.1) is reduced into the system of ODEs, then the eigenvalues of the linear part of the system are checked. As we know, a system is stable or asymptotically stable as real part of its eigenvalues are non-positive or negative, respectively.

Chapter 5 provides us with numerous illustrations and results obtained from three test problems with different parameters using the proposed method. As we shall see, for small values of diffusion coefficient $\hat{\gamma}$ the system converges to its equilibrium solutions $(\hat{\alpha}, \hat{\beta}/\hat{\alpha})$ if $\hat{\alpha}^2 - \hat{\beta} + 1 > 0$, otherwise the solution is unstable.

Finally, Chapter 6 ends with a brief conclusion.

CHAPTER 2

COMPACT FINITE DIFFERENCE METHOD

In this chapter, the Compact Finite Difference Method is elaborated on. Broadly speaking, the Compact Finite Difference formulation, or Hermitian formulation, is a numerical method to compute finite difference approximations. In the formulation of Compact Finite Difference approximation the influence of the derivatives of neighbouring points are included in the calculation. Although this approach causes an increase in accuracy it creates a global dependence (Lele, 1992), i.e., a change in the estimate of the derivative of the function at an arbitrary point affects the rest of the estimates of the derivatives. In other words, the compact schemes obtained are implicit meaning a system of equation is obtained when the scheme is applied to all the points and then solved simultaneously for all unknowns as we shall see (Hoffmann and Chiang, 2000).

2.1. The Construction of Compact Finite Difference Schemes for the First-Order Derivatives

Let the values of a function be given on set of points and let a uniformly spaced mesh to be considered and the points are indexed by i. The independent variable at the points is $x_i = a + h(i-1)$ for $1 \leq i \leq N$ where $h = (b-a)/(N-1)$ (provided that $[a,b]$ is the domain) and the function values at the nodes $f_i = f(x_i)$ are given. We start the formulation with a simple three-point formula of the first-order derivative using Hermitian formula given by (Hoffmann and Chiang, 2000)

$$H_i = \sum_{m=-1}^{m=+1} (a_m f_{i+m} + b_m f'_{i+m}) = 0 \tag{2.1}$$

or

$$a_{-1}f_{i-1} + a_0 f_i + a_1 f_{i+1} + b_{-1}f'_{i-1} + b_0 f'_i + b_1 f'_{i+1} = 0 \tag{2.2}$$

Expanding all the function values around the point x_i by Taylor series expansion Eq. (2.2) becomes

$$\begin{aligned}
&a_{-1}\left[f_i - hf_i' + \frac{1}{2!}h^2 f_i'' - \frac{1}{3!}h^3 f_i''' + \frac{1}{4!}h^4 f_i^{iv} - \frac{1}{5!}h^5 f_i^{v} + \frac{1}{6!}h^6 f_i^{vi} + \mathcal{O}(h^7)\right] + a_0 f_i +\\
&a_{1}\left[f_i + hf_i' + \frac{1}{2!}h^2 f_i'' + \frac{1}{3!}h^3 f_i''' + \frac{1}{4!}h^4 f_i^{iv} + \frac{1}{5!}h^5 f_i^{v} + \frac{1}{6!}h^6 f_i^{vi} + \mathcal{O}(h^7)\right] +\\
&b_{-1}\left[f_i' - hf_i'' + \frac{1}{2!}h^2 f_i''' - \frac{1}{3!}h^3 f_i^{iv} + \frac{1}{4!}h^4 f_i^{v} - \frac{1}{5!}h^5 f_i^{vi} + \mathcal{O}(h^6)\right] + b_0 f_i' +\\
&b_{1}\left[f_i' + hf_i'' + \frac{1}{2!}h^2 f_i''' + \frac{1}{3!}h^3 f_i^{iv} + \frac{1}{4!}h^4 f_i^{v} + \frac{1}{5!}h^5 f_i^{vi} + \mathcal{O}(h^6)\right] = 0
\end{aligned} \tag{2.3}$$

rewriting

$$\begin{aligned}
&(a_{-1} + a_0 + a_1)f_i + [h(a_1 - a_{-1}) + b_{-1} + b_0 + b_1]f_i' +\\
&\left[\frac{1}{2}h^2(a_1 + a_{-1}) + h(b_1 - b_{-1})\right] f_i'' + \left[\frac{1}{6}h^3(a_1 - a_{-1}) + \frac{1}{2}h^2(b_1 + b_{-1})\right] f_i''' +\\
&\left[\frac{1}{24}h^4(a_1 + a_{-1}) + \frac{1}{6}h^3(b_1 - b_{-1})\right] f_i^{iv} + \left[\frac{1}{120}h^5(a_1 - a_{-1}) + \frac{1}{24}h^4(b_1 + b_{-1})\right] f_i^{v} +\\
&\left[\frac{1}{720}h^6(a_1 + a_{-1}) + \frac{1}{120}h^5(b_1 - b_{-1})\right] f_i^{vi} = 0
\end{aligned} \tag{2.4}$$

In order for Eq. (2.4) to be exact all the coefficients need to be zero. However, only a few of them are set to zero, and the remaining higher order terms are truncated and formed a truncation error (TE). To obtain a third-order scheme, we set the coefficient of f_i, f_i', f_i'', and f_i''' equal to zeros.

$$a_{-1} + a_0 + a_1 = 0 \tag{2.5}$$

$$h(a_1 - a_{-1}) + b_{-1} + b_0 + b_1 = 0 \tag{2.6}$$

$$\frac{1}{2}h^2(a_1 + a_{-1}) + h(b_1 - b_{-1}) = 0 \tag{2.7}$$

$$\frac{1}{6}h^3(a_1 - a_{-1}) + \frac{1}{2}h^2(b_1 + b_{-1}) = 0 \tag{2.8}$$

Now, we have six unknowns with four equations, two of them are chosen freely, let those free parameters be b_1 and b_{-1}

$$a_1 = \frac{1}{2h}(-5b_1 - b_{-1}) \tag{2.9}$$

$$a_0 = \frac{2}{h}(b_1 - b_{-1}) \tag{2.10}$$

$$a_{-1} = \frac{1}{2h}(b_1 + 5b_{-1}) \tag{2.11}$$

$$b_0 = 2(b_1 + b_{-1}) \tag{2.12}$$

The Eq. (2.2) can be written as

$$\frac{a_{-1}}{b_1}f_{i-1} + \frac{a_0}{b_1}f_i + \frac{a_1}{b_1}f_{i+1} + \frac{b_{-1}}{b_1}f'_{i-1} + \frac{b_0}{b_1}f'_i + f'_{i+1} = 0 \tag{2.13}$$

where the equation is divided by b_1. The expressions given by (2.9 - 2.12) then should be redefined as

$$\frac{a_1}{b_1} = \frac{1}{2h}(-5 - \alpha) \tag{2.14}$$

$$\frac{a_0}{b_1} = \frac{2}{h}(1 - \alpha) \tag{2.15}$$

$$\frac{a_{-1}}{b_1} = \frac{1}{2h}(1 + 5\alpha) \tag{2.16}$$

$$\frac{b_0}{b_1} = 2(1 + \alpha) \tag{2.17}$$

where $\alpha = b_{-1}/b_1$. Substituting expressions (2.14 - 2.17) to Eq. (2.13) yields

$$\alpha f'_{i-1} + 2(1+\alpha)f'_i + f'_{i+1} = -\frac{1}{2h}(1+5\alpha)f_{1-i} - \frac{2}{h}(1-\alpha)f_i + \frac{1}{2h}(5+\alpha)f_{i+1} + TE \tag{2.18}$$

where

$$TE = \frac{1}{12}h^3(1-\alpha)f_i^{iv} + \frac{1}{60}h^4(1+\alpha)f_i^{v} + \frac{1}{180}h^5(1-\alpha)f_i^{vi} + \dots \tag{2.19}$$

For $\alpha = 1$ the first term of TE vanishes and the scheme becomes fourth-order, whereas, for other choices of α's the scheme is third-order. For $\alpha = 1$ the scheme is

$$f'_{i-1} + 4f'_i + f'_{i+1} = -\frac{3}{h}(f_{i-1} - f_{i+1}) \tag{2.20}$$

Now a general five-point formulation, also known as generalizations of Padé scheme, for the approximation of the first-order derivative can be expressed as (Lele, 1992)

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = a\frac{f_{i+1} - f_{i-1}}{2h} + b\frac{f_{i+2} - f_{i-2}}{4h} + c\frac{f_{i+3} - f_{i-3}}{6h} \tag{2.21}$$

The relation between the coefficients a, b, c and α, β are established by matching the Taylor series coefficients up to the needed order. The procedure is shown up to sixth-order schemes. To obtain higher order schemes the higher order terms of of Taylor series are retained.

The Taylor series expansions of the terms on the left-hand side of Eq. (2.21) are

$$f'_{i-2} = f'_i - 2hf''_i + \frac{4}{2!}h^2 f'''_i - \frac{8}{3!}h^3 f^{iv}_i + \frac{16}{4!}h^4 f^v_i \tag{2.22}$$

$$f'_{i-1} = f'_i - hf''_i + \frac{1}{2!}h^2 f'''_i - \frac{1}{3!}h^3 f^{iv}_i + \frac{1}{4!}h^4 f^v_i \tag{2.23}$$

$$f'_{i+1} = f'_i + hf''_i + \frac{1}{2!}h^2 f'''_i + \frac{1}{3!}h^3 f^{iv}_i + \frac{1}{4!}h^4 f^v_i \tag{2.24}$$

$$f'_{i+2} = f'_i + 2hf''_i + \frac{4}{2!}h^2 f'''_i + \frac{8}{3!}h^3 f^{iv}_i + \frac{16}{4!}h^4 f^v_i \tag{2.25}$$

Rewrite the left-hand side of Eq. (2.21) as

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = \beta(f'_{i-2} + f'_{i+2}) + \alpha(f'_{i-1} + f'_{i+1}) + f'_i \tag{2.26}$$

Substituting expressions (2.22 - 2.25) into Eq. (2.21) one gets

$$\begin{aligned} \beta\left[2f'_i + 4h^2 f'''_i + \frac{32h^4}{4!}f^v_i\right] + \alpha\left[2f'_i + h^2 f'''_i + \frac{2h^4}{4!}f^v_i\right] + f'_i = \\ (2\alpha + 2\beta + 1)f'_i + h^2(\alpha + 4\beta)f'''_i + \frac{2h^4}{4!}(\alpha + 16\beta)f^v_i. \end{aligned} \tag{2.27}$$

The Taylor series expansions of the terms on the right-hand side of Eq. (2.21) are

$$f_{i-3} = f_i - 3hf'_i + \frac{9}{2!}h^2 f''_i - \frac{27}{3!}h^3 f'''_i + \frac{81}{4!}h^4 f^{iv}_i - \frac{243}{4!}h^5 f^v_i \tag{2.28}$$

$$f_{i-2} = f_i - 2hf'_i + \frac{4}{2!}h^2 f''_i - \frac{8}{3!}h^3 f'''_i + \frac{16}{4!}h^4 f^{iv}_i - \frac{32}{4!}h^5 f^v_i \tag{2.29}$$

$$f_{i-1} = f_i - hf'_i + \frac{1}{2!}h^2 f''_i - \frac{1}{3!}h^3 f'''_i + \frac{1}{4!}h^4 f^{iv}_i - \frac{1}{4!}h^5 f^v_i \tag{2.30}$$

$$f_{i+1} = f_i + hf'_i + \frac{1}{2!}h^2 f''_i + \frac{1}{3!}h^3 f'''_i + \frac{1}{4!}h^4 f^{iv}_i + \frac{1}{4!}h^5 f^v_i \tag{2.31}$$

$$f_{i+2} = f_i + 2hf'_i + \frac{4}{2!}h^2 f''_i + \frac{8}{3!}h^3 f'''_i + \frac{16}{4!}h^4 f^{iv}_i + \frac{32}{4!}h^5 f^v_i \tag{2.32}$$

$$f_{i+3} = f_i + 3hf'_i + \frac{9}{2!}h^2 f''_i + \frac{27}{3!}h^3 f'''_i + \frac{81}{4!}h^4 f^{iv}_i + \frac{243}{4!}h^5 f^v_i \tag{2.33}$$

Expressions (2.28 - 2.33) are then substituted into the right-hand side of Eq. (2.21)

$$\begin{gathered}
\frac{a}{2h}(f_{i+1} - f_{i-1}) + \frac{b}{4h}(f_{i+2} - f_{i-2}) + \frac{c}{6h}(f_{i+3} - f_{i-3}) = \\
\frac{a}{2h}\left[2hf'_i + 2\frac{h^3}{3!}f'''_i + 2\frac{h^5}{5!}f^v_i\right] + \frac{b}{4h}\left[4hf'_i + 2\frac{8h^3}{3!}f'''_i + 2\frac{32h^5}{5!}f^v_i\right] + \frac{c}{6h}\left[6hf'_i + \right. \\
\left. 2\frac{27h^3}{3!}f'''_i + 2\frac{243h^5}{5!}f^v_i\right] = (a+b+c)f'_i + \frac{h^2}{3!}(a+4b+9c)f'''_i + \frac{h^4}{5!}(a+16b+81c)f^v_i
\end{gathered} \tag{2.34}$$

Combining expressions (2.27) and (2.34) one obtains

$$\begin{gathered}
(2\alpha + 2\beta{+}1)f'_i + h^2(\alpha + 4\beta)f'''_i + \frac{2h^4}{4!}(\alpha + 16\beta)f^v_i = \\
(a+b+c)f'_i + \frac{h^2}{3!}(a+4b+9c)f'''_i + \frac{h^4}{5!}(a+16b+81c)f^v_i
\end{gathered} \tag{2.35}$$

The relations among coefficients are established by equalizing the coefficients of the differentials

$$1 + 2\alpha + 2\beta = a + b + c \qquad \textit{Second-order} \tag{2.36}$$

$$2\frac{3!}{2!}(\alpha + 2^2\beta) = a + 2^2b + 3^2c \qquad \textit{Fourth-order} \tag{2.37}$$

$$2\frac{5!}{4!}(\alpha + 2^4\beta) = a + 2^4b + 3^4c \qquad \textit{Sixth-order} \tag{2.38}$$

$$2\frac{7!}{6!}(\alpha + 2^6\beta) = a + 2^6b + 3^6c \qquad \textit{Eighth-order} \tag{2.39}$$

$$2\frac{9!}{8!}(\alpha + 2^8\beta) = a + 2^8b + 3^8c \qquad \textit{Tenth-order} \tag{2.40}$$

The last two relations are written following the pattern of the previous ones, one can also expand the Taylor series for higher-order terms to see that it actually holds. To obtain a scheme of the desired order one uses the above relations up to the required order and the remaining terms in (2.35) are truncated, some chosen family of schemes and their corresponding truncation errors are listed in Table 2.1. Depending on the choice of β and α either a tridiagonal or a pentadiagonal schemes are generated. If $\beta = 0$ a variety of tridiagonal schemes are produced. For $\beta \neq 0$ pentadiagonal schemes are obtained. As for the number of stencils on the right-hand side, expression (2.21) gives stencil size of 3, 5 and 7 depending on the choice of b and c. For $b = c = 0$, the stencil size is 3, for $c = 0$, the stencil size is 5 and for $c \neq 0$, the stencil size is 7.

For example, for the fourth-order scheme the relations (2.36) and (2.37) are used, and the truncated terms are

$$TE = \left[\frac{1}{5!}(a + 2^4b + 3^4c) - \frac{2}{4!}(\alpha + 2^4\beta)\right] h^4 f_i^v. \tag{2.41}$$

Since the number of unknowns is five and the number of equations is two, three of the unknowns are chosen freely, so let $\beta = 0$, $c = 0$, and $a = \frac{2}{3}(\alpha + 2)$, $b = \frac{1}{3}(4\alpha - 1)$ with

$$TE = \frac{4}{5!}(3\alpha - 1)h^4 f_i^v. \tag{2.42}$$

With the values given above a one-parameter family (a.k.a α-family) of fourth-order tridiagonal systems can be produced. Notice that for $\alpha = \frac{1}{4}$, b vanishes and $a = \frac{3}{2}$, and one gets

$$\frac{1}{4}f'_{i-1} + f'_i + \frac{1}{4}f'_{i+1} = \frac{3}{2}\frac{f_{i+1} - f_{i-1}}{2h} \tag{2.43}$$

which is the same with Eq. (2.20). Another scheme that can be deduced from this family of schemes is by choosing $\alpha = \frac{1}{3}$. Notice that the first term of truncation error vanishes and $a = \frac{14}{9}$, $b = \frac{1}{9}$ so on gets a scheme of sixth-order accurate (Sari and Gürarslan, 2009; Sari et al., 2010; Gurarslan et al., 2013).

$$\frac{1}{3}f'_{i-1} + f'_i + \frac{1}{3}f'_{i+1} = \frac{14}{9}\frac{f_{i+1} - f_{i-1}}{2h} + \frac{1}{9}\frac{f_{i+2} - f_{i-2}}{4h} \tag{2.44}$$

2.2. The Construction of Compact Finite Difference Schemes for the Second-Order Derivatives

A similar approach to that of (2.21) can be used for the second-order derivative as follows (Hoffmann and Chiang, 2000)

$$\begin{aligned}\beta f''_{i-2} + \alpha f''_{i-1} + f''_i + \alpha f''_{i+1} + \beta f''_{i+2} &= \\ a\frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + b\frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2} &+ c\frac{f_{i+3} - 2f_i + f_{i-3}}{9h^2}.\end{aligned} \tag{2.45}$$

Following the same steps the constraints between the coefficients can be established as

$$1 + 2\alpha + 2\beta = a + b + c \qquad \textit{Second-order} \tag{2.46}$$

$$\frac{4!}{2!}(\alpha + 2^2\beta) = a + 2^2 b + 3^2 c \qquad \textit{Fourth-order} \tag{2.47}$$

$$\frac{6!}{4!}(\alpha + 2^4\beta) = a + 2^4 b + 3^4 c \qquad \textit{Sixth-order} \tag{2.48}$$

$$\frac{8!}{6!}(\alpha + 2^6\beta) = a + 2^6 b + 3^6 c \qquad \textit{Eighth-order} \tag{2.49}$$

$$\frac{10!}{8!}(\alpha + 2^8\beta) = a + 2^8 b + 3^8 c \qquad \textit{Tenth-order} \tag{2.50}$$

Similar way depending on the choice of α and β a tridiagonal or a pentadiagonal system can be generated. The number of stencils on the right-hand side can be seen from expression (2.45) that is for $b = c = 0$ the stencil size is 3, for $c = 0$, the stencil size is 5 and for $c \neq 0$ the stencil size is 7. Some selected family of schemes with the corresponding leading truncation error terms are presented in Table 2.2.

By choosing $\beta = 0$ and $c = 0$ an α-family of fourth-order of tridiagonal schemes is produced where $a = \frac{4}{3}(1 - \alpha)$, $b = \frac{1}{3}(-1 + 10\alpha)$ with truncation error

$$TE = \frac{4}{6!}(11\alpha - 2)h^4 f_i^{(5)} \tag{2.51}$$

setting $\alpha = \frac{1}{10}$ makes b disappear so one gets (Hoffmann and Chiang, 2000)

$$\frac{1}{10} f''_{i-1} + f''_i + \frac{1}{10} f''_{i+1} = \frac{6}{5} \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} \tag{2.52}$$

a fourth-order compact scheme. This unique choice of $\alpha = \frac{2}{11}$ makes the leading term of the truncation error vanish and one obtains a sixth-order scheme (Lele, 1992)

$$\frac{2}{11} f''_{i-1} + f''_i + \frac{2}{11} f''_{i+1} = \frac{12}{11} \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + \frac{3}{11} \frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2}. \tag{2.53}$$

Table 2.1. Selected compact finite difference schemes for the first-order derivative.

Eq.	α	β	a	b	c	Truncation error
2.37	α	0	$\frac{2}{3}(\alpha+2)$	$\frac{1}{3}(4\alpha-1)$	0	$\frac{4}{5!}(3\alpha-1)h^4 f_i^{(5)}$
2.38	α	0	$\frac{1}{6}(\alpha+9)$	$\frac{1}{15}(32\alpha-9)$	$\frac{1}{10}(-3\alpha+1)$	$\frac{12}{7!}(-8\alpha+3)h^6 f_i^{(7)}$
2.39	α	$\frac{1}{20}(-3+8\alpha)$	$\frac{1}{6}(12-7\alpha)$	$\frac{1}{150}(568\alpha-183)$	$\frac{1}{50}(9\alpha-4)$	$\frac{144}{9!}(2\alpha-1)h^8 f_i^{(9)}$
2.40	$\frac{1}{2}$	$\frac{1}{20}$	$\frac{17}{12}$	$\frac{101}{150}$	$\frac{1}{100}$	$\frac{144}{11!}h^{10} f_i^{(11)}$

Table 2.2. Selected compact finite difference schemes for the second-order derivative.

Eq.	α	β	a	b	c	Truncation error
2.47	α	0	$\frac{4}{3}(1-\alpha)$	$\frac{1}{3}(10\alpha-1)$	0	$\frac{4}{6!}(11\alpha-2)h^4 f_i^{(5)}$
2.48	α	β	$\frac{1}{4}(6-9\alpha-12\beta)$	$\frac{1}{5}(-3+24\alpha-6\beta)$	$\frac{1}{20}(2-11\alpha+124\beta)$	$-\frac{8}{8!}(9-38\alpha-214\beta)h^6 f_i^{(8)}$
2.49	α	$\frac{1}{214}(38\alpha-9)$	$\frac{1}{428}(696-1191\alpha)$	$\frac{1}{535}(245\alpha-294)$	$\frac{1}{2140}(1179\alpha-334)$	$\frac{1}{269640}(899\alpha-334)h^8 f_i^{(10)}$
2.50	$\frac{334}{899}$	$\frac{43}{1798}$	$\frac{1065}{1798}$	$\frac{1038}{899}$	$\frac{79}{1798}$	$\frac{619}{299043360}h^{10} f_i^{(12)}$

2.3. One-Sided Approximation

The schemes introduced so far are based on central difference approximations. These schemes will create difficulties for non-periodic boundary conditions. Therefore, a special type of schemes are to be introduced to overcome this problem. These schemes are called one-sided or non-central.

2.3.1. Boundary Formulation for the First-Order Derivative

To approximate the first-order derivative at the boundary node $i = 1$ the following relation can be used (Hoffmann and Chiang, 2000; Lele, 1992)

$$f_1' + \alpha f_2' = \frac{1}{h}(af_1 + bf_2 + cf_3 + df_4). \tag{2.54}$$

Expanding each function around the point x_1 by Taylor series and setting terms of various orders to zero gives the following relations between the coefficients

$$a + b + c + d = 0 \tag{2.55}$$

$$b + 2c + 3d = 1 + \alpha \qquad \textit{First-order} \tag{2.56}$$

$$b + 2^2c + 3^2d = 2\alpha \qquad \textit{Second-order} \tag{2.57}$$

$$b + 2^3c + 3^3d = 3\alpha \qquad \textit{Third-order} \tag{2.58}$$

$$b + 2^4c + 3^4d = 4\alpha \qquad \textit{Fourth-order} \tag{2.59}$$

Using equations above some selected schemes are presented in Table 2.3 with the corresponding leading order truncation error (on the right-hand side of Eq. (2.54)). The highest order that we can reach with Eq. (2.54) is fourth-order. To obtain higher order schemes one has to add more terms to the right-hand side (i.e., adding $ef_5 + \dots$).

For $\alpha = 0$ explicit schemes can be obtained. A third-order accurate explicit scheme is given by

$$f'_1 = \frac{1}{h}(-\frac{11}{6}f_1 + 3f_2 - \frac{3}{2}f_3 + \frac{1}{3}f_4) \tag{2.60}$$

with the leading order truncation error $\frac{-6}{4!}h^3 f_1^{(4)}$. A third-order accurate implicit scheme can be derived by choosing $\alpha = 2$

$$f'_1 + 2f'_2 = \frac{1}{h}(-\frac{5}{2}f_1 + 2f_2 + \frac{1}{2}f_3) \tag{2.61}$$

with the leading order truncation error $\frac{-2}{4!}h^3 f_1^{(4)}$. Notice that the truncation error of third-order explicit scheme is 3 times larger than that of the third-order implicit scheme.

For the boundary node i = N the following relation is used (Hoffmann and Chiang, 2000; Lele, 1992)

$$f'_N + \alpha f'_{N-1} = \frac{1}{h^2}(af_N + bf_{N-1} + cf_{N-2} + df_{N-3}) \tag{2.62}$$

The constraints among the coefficients are established in the same manner. Some group of schemes of different accuracy can be seen in Table 2.4. To obtain a third-order compact scheme set $\alpha = 2$

$$f'_N + 2f'_{N-1} = \frac{1}{h}(\frac{5}{2}f_N - 2f_{N-1} - \frac{1}{2}f_{N-2}) \tag{2.63}$$

For the same value of α expression (2.62) seems to give the same values of a, b, c and d as we obtained for boundary node $i = 1$ but with the opposite sign.

Table 2.3. Selected one-sided approximations for the boundary node $i = 1$ for the first-order derivative.

Eq.	α	a	b	c	d	Truncation error
2.57	α	$-\frac{1}{2}(3+\alpha+2d)$	$2+3d$	$-\frac{1}{2}(1-\alpha+6d)$	d	$\frac{1}{3!}(2-\alpha-6d)h^2 f_1^{(3)}$
2.58	α	$-\frac{1}{6}(11+2\alpha)$	$\frac{1}{2}(6-\alpha)$	$\frac{1}{2}(2\alpha-3)$	$\frac{1}{6}(2-\alpha)$	$\frac{2}{4!}(\alpha-3)h^3 f_1^{(4)}$
2.59	3	$-\frac{17}{6}$	$\frac{3}{2}$	$\frac{3}{2}$	$-\frac{1}{6}$	$\frac{6}{5!}h^4 f_1^{(5)}$

Table 2.4. Selected one-sided approximations for the boundary node $i = N$ for the first-order derivative.

Eq.	α	a	b	c	d	Truncation error
2.57	α	$-\frac{1}{2}(3+\alpha-2d)$	$-2+3d$	$\frac{1}{2}(1-\alpha-6d)$	d	$\frac{1}{3!}(\alpha-6d-2)h^2 f_1^{(3)}$
2.58	α	$\frac{1}{6}(11+2\alpha)$	$\frac{1}{2}(\alpha-6)$	$\frac{1}{2}(3-2\alpha)$	$\frac{1}{6}(\alpha-2)$	$\frac{2}{4!}(\alpha-3)h^3 f_1^{(4)}$
2.59	3	$\frac{17}{6}$	$-\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{6}$	$\frac{6}{5!}h^4 f_1^{(5)}$

2.3.2. Boundary Formulation for the Second-Order Derivative

For the boundary node $i = 1$ the following equation is used (Hoffmann and Chiang, 2000; Lele, 1992)

$$f''_1 + \alpha f''_2 = \frac{1}{h^2}(af_1 + bf_2 + cf_3 + df_4 + ef_5) \tag{2.64}$$

and for the boundary node $i = N$ the following equation is used

$$f''_N + \alpha f''_{N-1} = \frac{1}{h^2}(af_N + bf_{N-1} + cf_{N-2} + df_{N-3} + ef_{N-4}) \tag{2.65}$$

the relation between coefficients for both nodes is as follows

$$a + b + c + d + e = 0 \tag{2.66}$$

$$b + 2c + 3d + 4e = 0 \tag{2.67}$$

$$b + 2^2c + 3^2d + 4^2e = 2 + 2\alpha \qquad \textit{First-order} \tag{2.68}$$

$$b + 2^3c + 3^3d + 4^3e = 6\alpha \qquad \textit{Second-order} \tag{2.69}$$

$$b + 2^4c + 3^4d + 4^4e = 12\alpha \qquad \textit{Third-order} \tag{2.70}$$

$$b + 2^5c + 3^5d + 4^5e = 20\alpha \qquad \textit{Fourth-order} \tag{2.71}$$

A third-order accurate compact scheme can be obtained by choosing $\alpha = 11$, for the boundary node $i = 1$ the scheme is as follows

$$f''_1 + 11f''_2 = \frac{1}{h^2}(13f_1 - 27f_2 + 15f_3 - f_4) \tag{2.72}$$

and for the boundary node $i = N$

$$f''_N + 11f''_{N-1} = \frac{1}{h^2}(13f_N - 27f_{N-1} + 15f_{N-2} - f_{N-3}). \tag{2.73}$$

Observe that for both nodes the values of coefficients are the same. In Table 2.5 the relations are given for both nodes.

To approximate the derivative of a function with non-periodic boundary condition we need to combine one-sided schemes and schemes based on central differences. For the first-order derivative approximation let us bring back equations (2.61), (2.43), (2.44), (2.43) and (2.63), given respectively

$$f'_i + 2f'_{i+1} = \frac{1}{h}(-\frac{5}{2}f_i + 2f_{i+1} + \frac{1}{2}f_{i+2}), \quad i = 1$$

$$\frac{1}{4}f'_{i-1} + f'_i + \frac{1}{4}f'_{i+1} = \frac{3}{2}\frac{f_{i+1} - f_{i-1}}{2h}, \quad i = 2$$

$$\frac{1}{3}f'_{i-1} + f'_i + \frac{1}{3}f'_{i+1} = \frac{14}{9}\frac{f_{i+1} - f_{i-1}}{2h} + \frac{1}{9}\frac{f_{i+2} - f_{i-2}}{4h}, \quad i = 3, \ldots, N-2$$

$$\frac{1}{4}f'_{i-1} + f'_i + \frac{1}{4}f'_{i+1} = \frac{3}{2}\frac{f_{i+1} - f_{i-1}}{2h}, \quad i = N-1$$

$$f'_i + 2f'_{i-1} = \frac{1}{h}(\frac{5}{2}f_i - 2f_{i-1} - \frac{1}{2}f_{i-2}), \quad i = N$$

Notice that we used Eq. (2.63) twice for $i = 2$ and $i = N - 1$. The matrix form of the above system is

$$A_1 f' = \frac{1}{h}B_1 f \quad \Longrightarrow \quad f' = \frac{1}{h}A_1^{-1}B_1 f \quad \Longrightarrow \quad f' = \mathcal{D}^{(1)} f \tag{2.74}$$

where

$$A_1 = \begin{bmatrix} 1 & 2 & & & & & \\ \frac{1}{4} & 1 & \frac{1}{4} & & & & \\ & \frac{1}{3} & 1 & \frac{1}{3} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{1}{3} & 1 & \frac{1}{3} & \\ & & & & \frac{1}{4} & 1 & \frac{1}{4} \\ & & & & & 2 & 1 \end{bmatrix} \tag{2.75}$$

and

$$B_1 = \begin{bmatrix} -\frac{5}{2} & 2 & \frac{1}{2} & & & & & \\ -\frac{3}{4} & 0 & \frac{3}{4} & & & & & \\ -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{36} & -\frac{7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ & & & & -\frac{3}{4} & 0 & \frac{3}{4} \\ & & & & -\frac{1}{2} & -2 & \frac{5}{2} \end{bmatrix} \tag{2.76}$$

Similarly, to approximate second-order derivative we use equations (2.72), (2.52), (2.53), (2.52) and (2.73), given respectively

$$f_i'' + 11f_{i+1}'' = \frac{1}{h^2}(13f_i - 27f_{i+1} + 15f_{i+2} - f_{i+3}), \quad i = 1$$

$$\frac{1}{10}f_{i-1}'' + f_i'' + \frac{1}{10}f_{i+1}'' = \frac{6}{5}\frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}, \quad i = 2$$

$$\frac{2}{11}f_{i-1}'' + f_i'' + \frac{2}{11}f_{i+1}'' = \frac{3}{11}\frac{f_{i-2} - 2f_i + f_{i+2}}{4h^2} + \frac{12}{11}\frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}, \quad i = 3, \ldots, N-2$$

$$\frac{1}{10}f_{i-1}'' + f_i'' + \frac{1}{10}f_{i+1}'' = \frac{6}{5}\frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}, \quad i = N-1$$

$$11f_{i-1}'' + f_i'' = \frac{1}{h^2}(13f_i - 27f_{i-1} + 15f_{i-2} - f_{i-3}), \quad i = N$$

The matrix form of the above system is

$$A_2 f'' = \frac{1}{h^2} B_2 f \quad \Longrightarrow \quad f'' = \frac{1}{h^2} A_2^{-1} B_2 f \quad \Longrightarrow \quad f'' = \mathcal{D}^{(2)} f \tag{2.77}$$

where

$$A_2 = \begin{bmatrix} 1 & 11 & & & & & \\ \frac{1}{10} & 1 & \frac{1}{10} & & & & \\ & \frac{2}{11} & 1 & \frac{2}{11} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{2}{11} & 1 & \frac{2}{11} & \\ & & & & \frac{1}{10} & 1 & \frac{1}{10} \\ & & & & & 11 & 1 \end{bmatrix} \tag{2.78}$$

and

$$B_2 = \begin{bmatrix} 13 & -27 & 15 & -1 & & & & \\ \frac{6}{5} & -\frac{12}{5} & \frac{6}{5} & & & & & \\ \frac{3}{44} & \frac{12}{11} & -\frac{51}{22} & \frac{12}{11} & \frac{3}{44} & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \frac{3}{44} & \frac{12}{11} & -\frac{51}{22} & \frac{12}{11} & \frac{3}{44} & \\ & & & & \frac{6}{5} & -\frac{12}{5} & \frac{6}{5} & \\ & & & 13 & -27 & 15 & -1 & \end{bmatrix} \tag{2.79}$$

Table 2.5. Selected one-sided approximations for the boundary node $i = 1$ and $i = N$ for the second-order derivative.

Eq.	α	a	b	c	d	e	Truncation error
2.57	α	$\alpha + 2 + e$	$-(2\alpha+5+4e)$	$\alpha + 4 + 6e$	$-(1 + 4e)$	e	$\frac{1}{12}(\alpha + 12e - 11)h^2 f_1^{(4)}$
2.58	α	$\frac{1}{12}(11\alpha + 35)$	$-\frac{1}{3}(5\alpha + 26)$	$\frac{1}{2}(\alpha + 19)$	$\frac{1}{3}(\alpha - 14)$	$\frac{1}{12}(11 - \alpha)$	$\frac{1}{12}(\alpha - 10)h^3 f_1^{(5)}$
2.59	10	$\frac{145}{12}$	$-\frac{76}{3}$	$\frac{29}{2}$	$-\frac{4}{3}$	$\frac{1}{12}$	$\frac{7}{180}h^4 f_1^{(6)}$

CHAPTER 3

ADAPTIVE STEP-SIZE RUNGE-KUTTA METHOD

In this chapter, a brief introduction to an adaptive step-size Runge-Kutta(RK) method is to be given. The step-size adjustment method that is of interest is based on a large family of Runge-Kutta methods, originally found by Erwin Fehlberg (Fehlberg, 1968, 1969). The advantage of the method Fehlberg presented is that it uses embedded methods that is the identical function evaluations are used in conjunction with each other to create methods of different order and similar error constants. In other words, an embedded method from the family of RK provides us with two ordinary RK formulas of different order sharing the same evaluation points. The difference between these two estimates can be used as an estimate of the local truncation error to adjust the step-size.

3.1. A Fifth-order Adaptive Step-size Runge-Kutta Method

In this section, a fifth-order adaptive step-size RK method with an error estimator of order sixth is to be presented. This adaptive time integrator is based on embedded RK formulas by (Dormand and Prince, 1980; Prince and Dormand, 1981). The estimate of local truncation error together with the predefined error tolerance allows the step-size to be determined automatically. Given an ordinary differential equation

$$\frac{du}{dt} = F(t, u), \quad u(t_0) = u_0 \tag{3.1}$$

where in the case of system of ordinary differential equations u and F represent vectors. The general m-stage RK formula for the approximations u_n to the exact solution $u(t_n)$ at the point t_n, where $t_{n+1} = t_n + \Delta t_n$ for $n = 0, 1, 2, \ldots$ with Δt_n as integration step-size, can be given by

$$u_{n+1} = u_n + \Delta t_n \sum_{s=1}^{m} \xi_s k_s \tag{3.2}$$

where

$$
\begin{aligned}
k_1 &= F(t_n, u_n) \\
k_s &= F(t_n + \lambda_s \Delta t_n, u_n + \Delta t_n \sum_{r=1}^{s-1} \mu_{r,s} k_r), \quad (s = 2, 3, \ldots, m)
\end{aligned}
\tag{3.3}
$$

where k_s stands for the approximated slope (İmamoğlu Karabaş et al., 2022; Cicek et al., 2022; Bahar and Gurarslan, 2020). A 7-stage explicit RK formula of fifth order can be obtained using coefficients λ_s, $\mu_{r,s}$, $\tilde{\xi}$ and ξ_s given in Table 3.1 (Dormand and Prince, 1980). To control the step-size a lower-order method, i.e., a 6-stage explicit RK formula of fourth order is derived using coefficients from the same table to obtain an estimate of local truncation error. The 7-stage explicit fifth-order RK formula is given by

$$
u_{n+1} = u_n + \Delta t_n \sum_{s=1}^{7} \xi_s k_s + \mathcal{O}(\Delta t^6). \tag{3.4}
$$

and the embedded fourth-order formula can be obtained by

$$
u^*_{n+1} = u_n + \Delta t_n \sum_{s=1}^{6} \tilde{\xi}_s k_s + \mathcal{O}(\Delta t^5) \tag{3.5}
$$

The difference between (3.4) and (3.5) gives an estimate of local truncation error

$$
e_n = u_{n+1} - u^*_{n+1} = \Delta t_n \sum_{s=1}^{7} (\xi_s - \tilde{\xi}_s) k_s \tag{3.6}
$$

The goal is to keep $|e_n|$ (in the vector case, a norm) less than some predetermined error tolerance ϵ, i.e., the solution satisfying $|e_n| < \epsilon$ is accepted otherwise it is rejected. To construct the formula for selecting the new step-size the local truncation errors of (3.4) and (3.5) are of use

$$u(t_n) - u_n^* = C^* \Delta t_n^5 + \dots$$

$$u(t_n) - u_n = C \Delta t_n^6 + \dots$$

where C^* and C are unknown constants. From equations above one gets

$$u_n - u_n^* = C^* \Delta t_n^5 + \dots$$

Therefore

$$|e_n| \approx |C^*| \Delta t_n^5 \tag{3.7}$$

Regardless of whether the condition $|e_n| < \epsilon$ satisfies or not, the step-size should be chosen just to meet the error tolerance. To do so, let the new step-size be $\tilde{\Delta} t_n$, then trivially one gets

$$|e_n| \approx |C^*| \tilde{\Delta} t_n^5 < \epsilon \tag{3.8}$$

substituting $|C^*|$ from (3.7) into (3.8) gives the new step in time as follows

$$\tilde{\Delta} t_n < \Delta t_n \left| \frac{\epsilon}{e_n} \right|^{1/5} \tag{3.9}$$

a safety factor S (a few percentage smaller then one) is multiplied to the right-hand side of (3.9) to satisfy the inequality, thus the step-size controller is as follows

$$\tilde{\Delta} t_n := S \Delta t_n \left| \frac{\epsilon}{e_n} \right|^{1/5}. \tag{3.10}$$

The Eq. (3.10) keeps the step-size within the desired accuracy ϵ. As we mentioned above the step-size is updated for every step whether the step is accepted or rejected. In the case it is accepted ($|e_n| < \epsilon$), because of the possibility of it being too small the equation determines how much it can be safely increased for the next step (Press and Teukolsky, 1992). If the step is failed ($|e_n| > \epsilon$) then the equation determines how much to decrease the step size and retry the failed step.

Table 3.1. Butcher array for explicit embedded RK formulas.

λ_s	$\mu_{r,s}$						$\tilde{\xi}_s$	ξ_s
0							$\frac{35}{384}$	$\frac{5179}{57600}$
$\frac{1}{5}$	$\frac{1}{5}$						0	0
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					$\frac{500}{1113}$	$\frac{7571}{16695}$
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				$\frac{125}{192}$	$\frac{393}{640}$
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			$-\frac{2187}{6784}$	$-\frac{92097}{339200}$
1	$\frac{9017}{3168}$	$\frac{-355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		$\frac{11}{84}$	$\frac{187}{2100}$
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	$\frac{1}{40}$

CHAPTER 4

STABILITY ANALYSIS

In this chapter, the analytical stability of diffusion-free Brusselator model is checked along with the stability of semi-discrete reaction-diffusion Brusselator model, the latter proves the stability of the proposed method. To analyze this we have used definitions and theorems from stability theory of ODEs.

Let us consider the autonomous system of ordinary differential equation of the form

$$\frac{dx}{dt} = f(x) \tag{4.1}$$

Then, x_0 is called the critical point, or equilibrium solution, of the above system if $f(x_0) = 0$.

Theorem 4.1 *Consider a non-linear system of the form* $\frac{d\mathcal{W}}{dt} = A\mathcal{W} + F(\mathcal{W})$ *and the related linear system of this is* $\frac{d\mathcal{W}}{dt} = A\mathcal{W}$*. Let* ***0*** *be a simple critical point of the non-linear system and*

$$\|F(\mathcal{W})\|/\|\mathcal{W}\| \to 0 \quad as \quad \mathcal{W} \to \boldsymbol{0}.$$

If the critical point ***0*** *of linear system is asymptotically stable, then the critical point* ***0*** *of non-linear system is also asymptotically stable.*

Proof For proof see (Browder et al., 1998).

Now, let us consider the one-dimensional Brusselator model

$$\begin{aligned}\frac{\partial u}{\partial t} &= \hat{\alpha} - (\hat{\beta} + 1)u + u^2 v + \hat{\gamma}\frac{\partial^2 u}{\partial x^2} \\ \frac{\partial v}{\partial t} &= \hat{\beta} u - u^2 v + \hat{\gamma}\frac{\partial^2 v}{\partial x^2}\end{aligned} \tag{4.2}$$

Neglecting the diffusion term and by $du/dt = dv/dt = 0$ the critical points, or equilibrium solutions, are $u = \hat{\alpha}$ and $v = \frac{\hat{\beta}}{\hat{\alpha}}$. In order to apply Theorem 4.1 the critical points need to be shifted to zeros, therefore let us substitute

$$\mathcal{U} = u - \hat{\alpha} \quad and \quad \mathcal{V} = v - \frac{\hat{\beta}}{\hat{\alpha}}$$

into the Eq. (4.2) then the following system is obtained

$$\begin{aligned} \frac{\partial \mathcal{U}}{\partial t} &= (\hat{\beta} - 1)\mathcal{U} + \hat{\alpha}^2\mathcal{V} + \mathcal{U}^2\mathcal{V} + 2\hat{\alpha}\mathcal{U}\mathcal{V} + \frac{\hat{\beta}}{\hat{\alpha}}\mathcal{U}^2 \\ \frac{\partial \mathcal{V}}{\partial t} &= -\hat{\beta}\mathcal{U} - \hat{\alpha}^2\mathcal{V} - \mathcal{U}^2\mathcal{V} - 2\hat{\alpha}\mathcal{U}\mathcal{V} - \frac{\hat{\beta}}{\hat{\alpha}}\mathcal{U}^2. \end{aligned} \tag{4.3}$$

Eq. (4.3) is of the form

$$\frac{d}{dt}\begin{bmatrix}\mathcal{U}\\ \mathcal{V}\end{bmatrix} = \underbrace{\begin{bmatrix}(\hat{\beta}-1) & \hat{\alpha}^2 \\ -\hat{\beta} & -\hat{\alpha}^2\end{bmatrix}}_{\mathfrak{C}}\begin{bmatrix}\mathcal{U}\\ \mathcal{V}\end{bmatrix} + \begin{bmatrix}\mathcal{U}^2\mathcal{V} + 2\hat{\alpha}\mathcal{U}\mathcal{V} + \frac{\hat{\beta}}{\hat{\alpha}}\mathcal{U}^2 \\ -\mathcal{U}^2\mathcal{V} - 2\hat{\alpha}\mathcal{U}\mathcal{V} - \frac{\hat{\beta}}{\hat{\alpha}}\mathcal{U}^2\end{bmatrix} \tag{4.4}$$

The eigenvalues of matrix $\mathfrak{C}$ are

$$\lambda_{1,2} = \frac{-(\hat{\alpha}^2 - \hat{\beta} + 1) \pm \sqrt{(\hat{\alpha}^2 - \hat{\beta} + 1)^2 - 4\hat{\alpha}^2}}{2}$$

and by Euclidean norm the following condition also holds

$$\|F(\mathcal{W})\|/\|\mathcal{W}\| \to 0 \quad as \quad \mathcal{W} \to \mathbf{0}.$$

The bifurcation diagram shows that $Re(\lambda) < 0$ when $\hat{\alpha}^2 - \hat{\beta} + 1 > 0$ (Twizell et al., 1999), thus $\mathcal{U} \to 0(u \to \hat{\alpha})$ and $\mathcal{V} \to 0(v \to \hat{\beta}/\hat{\alpha})$ as time increases and the system does not converge to its critical points whenever the condition does not satisfy. However, the main concern is what happens when the diffusion term is inserted, given as follows

$$\begin{aligned} \frac{\partial \mathcal{U}}{\partial t} &= (\hat{\beta}-1)\mathcal{U} + \hat{\alpha}^2\mathcal{V} + \mathcal{U}^2\mathcal{V} + 2\hat{\alpha}\mathcal{U}\mathcal{V} + \frac{\hat{\beta}}{\hat{\alpha}}\mathcal{U}^2 + \hat{\gamma}\frac{\partial^2 \mathcal{U}}{\partial x^2} \\ \frac{\partial \mathcal{V}}{\partial t} &= -\hat{\beta}\mathcal{U} - \hat{\alpha}^2\mathcal{V} - \mathcal{U}^2\mathcal{V} - 2\hat{\alpha}\mathcal{U}\mathcal{V} - \frac{\hat{\beta}}{\hat{\alpha}}\mathcal{U}^2 + \hat{\gamma}\frac{\partial^2 \mathcal{V}}{\partial x^2} \end{aligned} \tag{4.5}$$

After spatial discretization the semi-discrete form of Eq. (4.5) takes the following form

$$\frac{d}{dt}\begin{bmatrix} \tilde{\mathcal{U}} \\ \tilde{\mathcal{V}} \end{bmatrix} = \begin{bmatrix} (\hat{\beta}-1)\mathcal{I} + \hat{\gamma}\mathcal{D}^{(2)} & \hat{\alpha}^2\mathcal{I} \\ -\hat{\beta}\mathcal{I} & -\hat{\alpha}^2\mathcal{I} + \hat{\gamma}\mathcal{D}^{(2)} \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{U}} \\ \tilde{\mathcal{V}} \end{bmatrix} + \begin{bmatrix} \tilde{\mathcal{U}}^2\tilde{\mathcal{V}} + 2\hat{\alpha}\tilde{\mathcal{U}}\tilde{\mathcal{V}} + \frac{\hat{\beta}}{\hat{\alpha}}\tilde{\mathcal{U}}^2 \\ -\tilde{\mathcal{U}}^2\tilde{\mathcal{V}} - 2\hat{\alpha}\tilde{\mathcal{U}}\tilde{\mathcal{V}} - \frac{\hat{\beta}}{\hat{\alpha}}\tilde{\mathcal{U}}^2 \end{bmatrix} \tag{4.6}$$

where $\tilde{\mathcal{U}} = \left[\mathcal{U}_1, \mathcal{U}_2, \ldots \mathcal{U}_N\right]^T$, $\tilde{\mathcal{V}} = \left[\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N\right]^T$, $\mathcal{I}_{N\times N}$ is the identity matrix and $\mathcal{D}^{(2)}_{N\times N}$ is the constant matrix to approximate second-order spatial derivative. The above equation can be expressed as follows

$$\frac{d\tilde{\mathcal{W}}}{dt} = A\tilde{\mathcal{W}} + F(\tilde{\mathcal{W}}) \tag{4.7}$$

where $\tilde{\mathcal{W}} = \left[\tilde{\mathcal{U}}, \tilde{\mathcal{V}}\right]^T$, $A = \begin{bmatrix} (\hat{\beta}-1)\mathcal{I} + \hat{\gamma}\mathcal{D}^{(2)} & \hat{\alpha}^2\mathcal{I} \\ -\hat{\beta}\mathcal{I} & -\hat{\alpha}^2\mathcal{I} + \hat{\gamma}\mathcal{D}^{(2)} \end{bmatrix}_{2N\times 2N}$ and

$$F(\tilde{\mathcal{W}}) = \begin{bmatrix} \tilde{\mathcal{U}}^2\tilde{\mathcal{V}} + 2\hat{\alpha}\tilde{\mathcal{U}}\tilde{\mathcal{V}} + \frac{\hat{\beta}}{\hat{\alpha}}\tilde{\mathcal{U}}^2 \\ -\tilde{\mathcal{U}}^2\tilde{\mathcal{V}} - 2\hat{\alpha}\tilde{\mathcal{U}}\tilde{\mathcal{V}} - \frac{\hat{\beta}}{\hat{\alpha}}\tilde{\mathcal{U}}^2 \end{bmatrix}_{2N\times N}.$$

Again by Theorem 4.1, the non-linear system (4.7) is stable if the related linear system is stable in addition to the following condition

$$\|F(\tilde{\mathcal{W}})\|/\|\tilde{\mathcal{W}}\| \to 0 \quad as \quad \tilde{\mathcal{W}} \to 0.$$

Again applying Euclidean norm one gets

$$\frac{\sqrt{2}\big[(\mathcal{U}_1^2\mathcal{V}_1+2\hat{\alpha}\mathcal{U}_1\mathcal{V}_1+(\hat{\beta}/\hat{\alpha})\mathcal{U}_1^2)^2+\cdots+(\mathcal{U}_N^2\mathcal{V}_N+2\hat{\alpha}\mathcal{U}_N\mathcal{V}_N+(\hat{\beta}/\hat{\alpha})\mathcal{U}_N^2)^2\big]^{1/2}}{\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2}}$$

and proving that the following expression goes to zero is sufficient

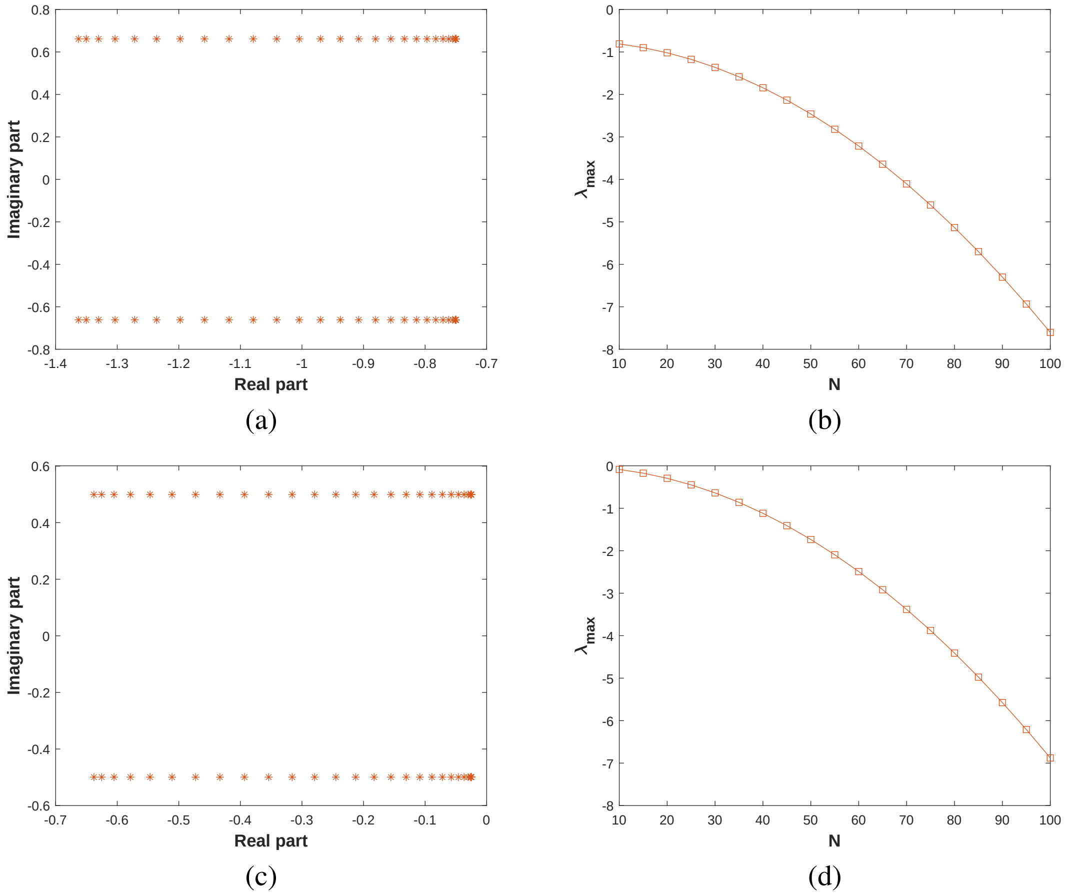
$$\frac{|\mathcal{U}_i^2\mathcal{V}_i+2\hat{\alpha}\mathcal{U}_i\mathcal{V}_i+(\hat{\beta}/\hat{\alpha})\mathcal{U}_i^2|}{\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2}}$$

the proof continues as follows

$$\begin{aligned}
&\leq \frac{|\mathcal{U}_i^2\mathcal{V}_i|+2\hat{\alpha}|\mathcal{U}_i\mathcal{V}_i|+(\hat{\beta}/\hat{\alpha})|\mathcal{U}_i^2|}{\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2}} \\
&\leq \frac{|\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2||\mathcal{V}_i|}{\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2}} \\
&\quad + \frac{2\hat{\alpha}|\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2|}{\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2}} \\
&\quad + \frac{(\hat{\beta}/\hat{\alpha})|\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2|}{\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2}} \\
&= (|\mathcal{V}_i|+2\hat{\alpha}+\hat{\beta}/\hat{\alpha})\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2} \to 0
\end{aligned}$$

as $\big[\mathcal{U}_1^2+\mathcal{U}_2^2+\cdots+\mathcal{U}_N^2+\mathcal{V}_1^2+\mathcal{V}_2^2+\cdots+\mathcal{V}_N^2\big]^{1/2} \to 0$. Hence, the condition holds. Therefore, the stability of non-linear system (4.7) depends on the stability of the related linear system

$$\frac{d\tilde{\mathcal{W}}}{dt}=A\tilde{\mathcal{W}}$$



(a)

(b)

(c)

(d)

Figure 4.1. For $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.0001$ 4.1a gives the eigenvalues where $N = 30$ and 4.1b gives the maximum eigenvalue as N increases. Similarly for $\hat{\alpha} = 0.5$, $\hat{\beta} = 1.2$ and $\hat{\gamma} = 0.0001$ in 4.1c and 4.1d,respectively.

The stability of equation above depends on the eigenvalues of the matrix A. The system is stable or asymptotically stable only if the real part of the eigenvalues are non-positive or negative, respectively. Figure 4.1 exhibits the eigenvalues of A for different parameters. We can see for small values of $\hat{\gamma}$ and the condition $\hat{\alpha}^2 - \hat{\beta} + 1 > 0$ the eigenvalues are negative for our proposed method. Hence, our algorithm is stable.

CHAPTER 5

NUMERICAL SIMULATIONS

In this section, numerical solutions of the Brusellator system is estimated for three problems to test the performance and efficiency of the proposed numerical schemes. One-dimensional and two-dimensional cases of the model are considered to obtain different types of patterns shown in Figures 5.1 - 5.10 supported with tables of values. For Problem 5.3 we have the exact solution to demonstrate the effectiveness and accuracy of the method. Tables 5.3 - 5.7 provides us with different types of results, of which some are compared to results available in the literature. The numerical computation is performed using uniform grids. All computations were done by codes produced in MatLab.

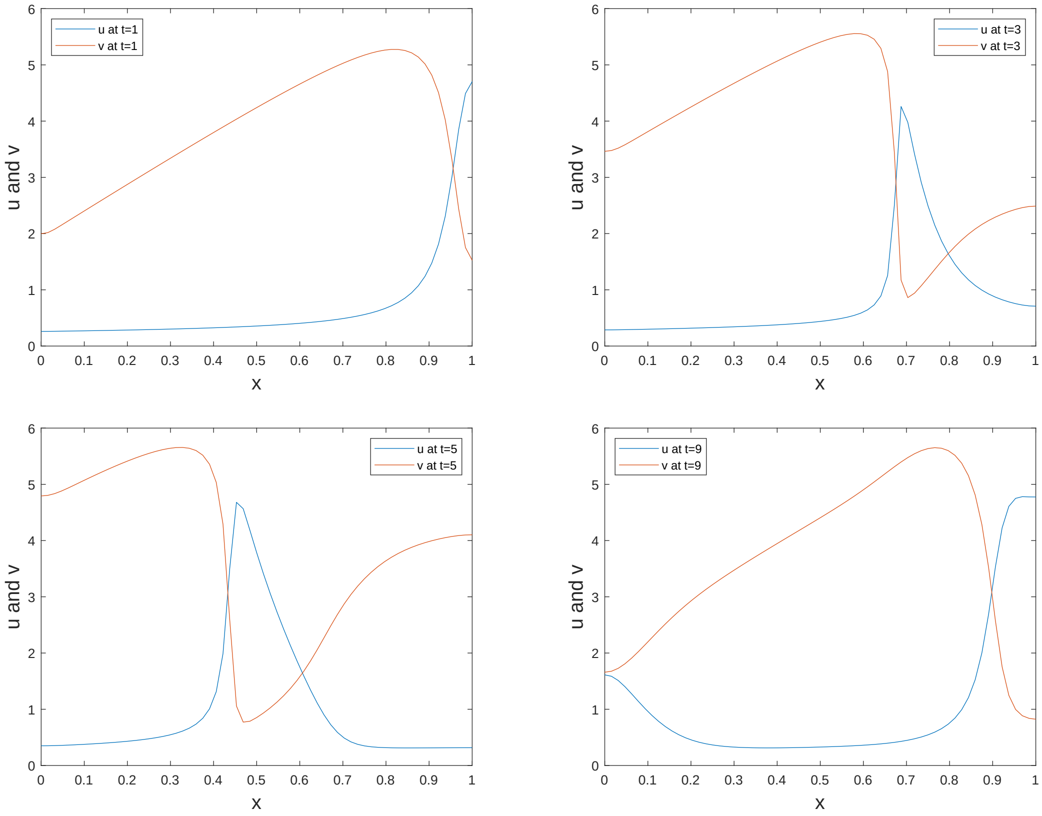


Figure 5.1. Physical behaviour of two species u and v at different times for Problem 5.1 for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.0001$.

Table 5.1. Concentration values for oscillatory behaviour of Problem 5.1.

Species	t	$x = 0.0$	$x = 0.2$	$x = 0.4$	$x = 0.6$	$x = 0.8$	$x = 1.0$
u	5	0.354180	0.430663	1.174895	1.736710	0.316833	0.315900
	12.79	0.357934	0.430911	1.179657	1.644320	0.317940	0.315145
	7	0.577109	3.890935	0.963604	0.319690	0.362983	0.406904
	14.79	0.613894	3.893757	0.962021	0.318195	0.361052	0.403287
v	5	4.849362	5.412711	5.184037	1.584278	3.640803	4.091804
	12.79	4.880870	5.410437	5.180623	1.642583	3.613300	4.062906
	7	5.758697	0.835523	2.196140	3.600124	4.929378	5.293889
	14.79	5.735633	0.835052	2.197824	3.635534	4.907729	5.271856

Problem 5.1 *Consider one-dimensional Brusselator system (Alqahtani, 2018; Kumar et al., 2019)*

$$\begin{aligned} \frac{\partial u}{\partial t} &= \hat{\alpha} - (\hat{\beta}+1)u + u^2 v + \hat{\gamma}\frac{\partial^2 u}{\partial x^2} \\ \frac{\partial v}{\partial t} &= \hat{\beta}u - u^2 v + \hat{\gamma}\frac{\partial^2 v}{\partial x^2} \end{aligned} \tag{5.1}$$

over the domain $[0, 1]$ *with initial conditions*

$$u(x,0) = 0.5, \quad v(x,0) = 1 + 5x \tag{5.2}$$

and Neumann boundary conditions

$$u_x(0,t) = u_x(1,t) = v_x(0,t) = v_x(1,t) = 0. \tag{5.3}$$

The simulations for Problem 5.1 are obtained by dividing the given domain $[0, 1]$ into 64 subintervals ($N = 64$). For time integration a predetermined accuracy tolerance of $\epsilon = 10^{-4}$ is used, see Section 3.1. Choosing different parameters $\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$ different types

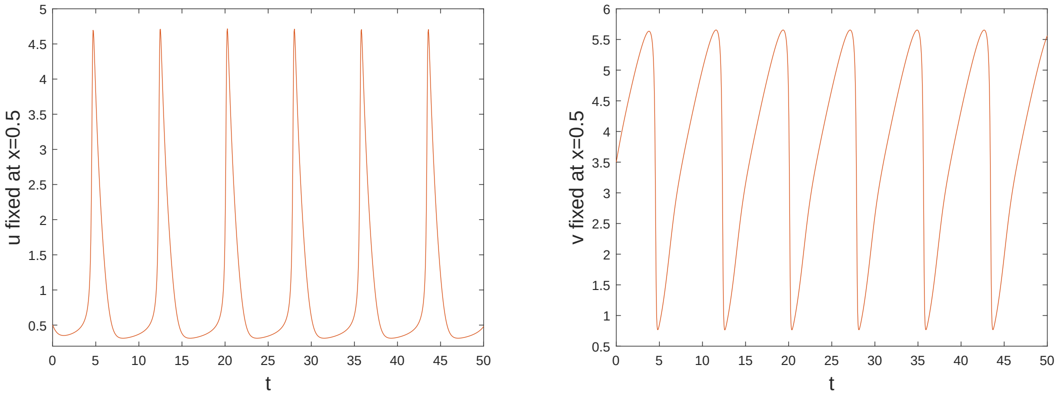


Figure 5.2. The solution profile of u and v of Problem 5.1 fixed at $(x, y) = (0.5, 0.5)$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$.

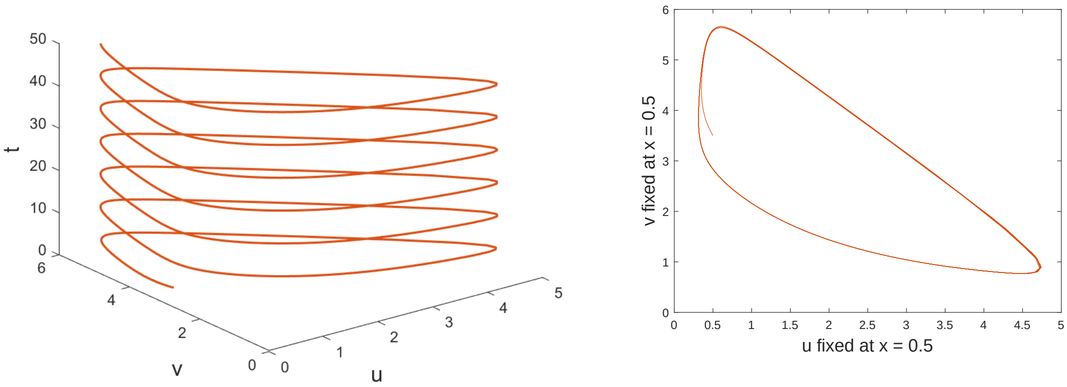


Figure 5.3. The path taken by (u, v) as time increase (left) and the (u, v)-plane (right) of Problem 5.1 fixed at $x = 0.5$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.0001$.

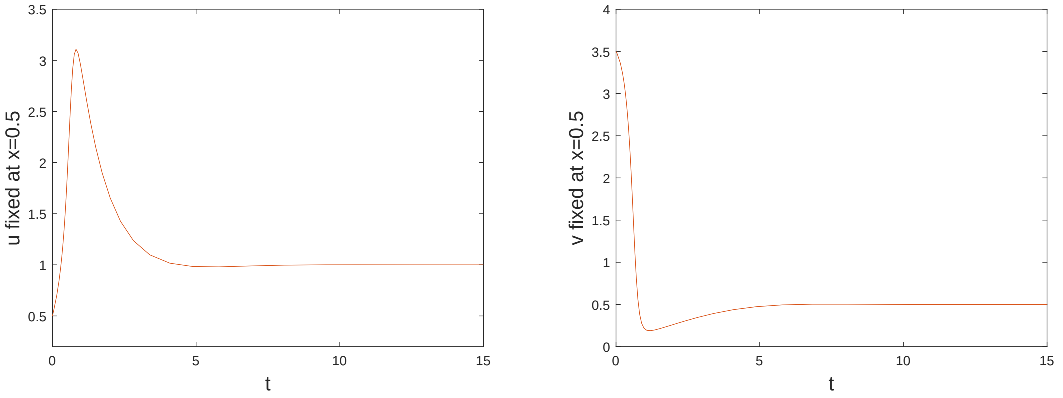


Figure 5.4. The solution profile of u and v of Problem 5.1 fixed at $x = 0.5$ over $0 < t \leq 15$ for $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.0001$.

of patterns can be captured. In Figure 5.1 with parameters chosen as $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.0001$ the interaction between two species u and v at different times are shown. For this particular set of parameters we have $\hat{\alpha}^2 - \hat{\beta} + 1 < 0$ which explains why the solution is oscillatory, see Figure 5.2. Additionally, the oscillatory behaviour of solution can be verified by values given in Table 5.1. According to Table 5.1 one period of u and v is estimated to be 7.79 seconds. For the parameters $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.0001$ patterns are exhibited in Figure 5.4 which seem to converge to the equilibrium points $(\hat{\alpha}, \hat{\beta}/\hat{\alpha}) = (1, 0.5)$ since $\hat{\alpha}^2 - \hat{\beta} + 1 > 0$.

Problem 5.2 *Consider the two dimensional Brusselator system with the initial data (Alqahtani, 2018; Kumar et al., 2019)*

$$u(x, y, 0) = 0.5 + y, \quad (x, y) \in [0, 1] \times [0, 1]$$

$$v(x, y, 0) = 1 + 5x, \quad (x, y) \in [0, 1] \times [0, 1]$$

and Neumann boundary conditions

$$\frac{\partial u}{\partial n} = \frac{\partial v}{\partial n} = 0, \quad (x, y, t) \in [0, 1] \times [0, 1] \times (0, T]$$

Table 5.2. The values of u and v converges to $\hat{\alpha}$ and $\hat{\beta}/\hat{\alpha}$, resp., as time increases.

t	(0.2,0.4)		(0.4,0.6)		(0.6,0.8)	
	u	v	u	v	u	v
1.0	2.51576	0.23445	2.79694	0.16647	3.13338	0.14974
2.0	1.51809	0.28672	1.59556	0.27191	1.71789	0.25475
4.0	1.00981	0.44443	1.01620	0.43831	1.02830	0.42947
6.0	0.98342	0.49924	0.98225	0.49841	0.98119	0.49682
8.0	0.99745	0.50271	0.99702	0.50294	0.99640	0.50314
10.0	1.00053	0.50034	1.00055	0.50040	1.00054	0.50050
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
∞	1.0	0.5	1.0	0.5	1.0	0.5

In Problem 5.2 the illustrations are obtained by choosing a 20×20 spatial mesh grid and $\epsilon = 10^{-4}$ for time integration. With $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.0001$ the solution profile is exhibited in Figure 5.5 and it converges to the equilibrium points $(\hat{\alpha}, \hat{\beta}/\hat{\alpha}) = (1, 0.5)$ since $\hat{\alpha}^2 - \hat{\beta} + 1 > 0$, additionally the values in Table 5.2 also demonstrates the convergence. Different patterns of concentration of v and its contour form are plotted in Figure 5.6 with parameters $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$ at different times $t = 5, 7, 9, 11.5, 12.2, 14.2, 15$ and it can be observed that the pattern has an oscillatory behaviour, i.e., the pattern at $t = 5$ is almost the same with the pattern at $t = 12.2$. Figure 5.7 exhibits the oscillatory behavior of the solution profile of u and v while Figure 5.8 shows the path taken by (u, v) as time increases (left) and the (u, v)-plane (right) fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 50$, with the same parameters.

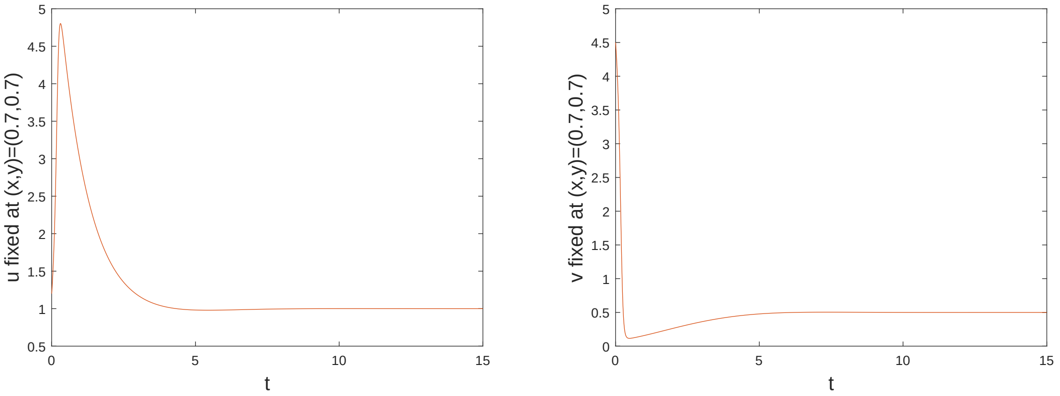


Figure 5.5. The solution profile of u and v of Problem 5.2 fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 15$ for $\hat{\alpha} = 1$, $\hat{\beta} = 0.5$ and $\hat{\gamma} = 0.002$.

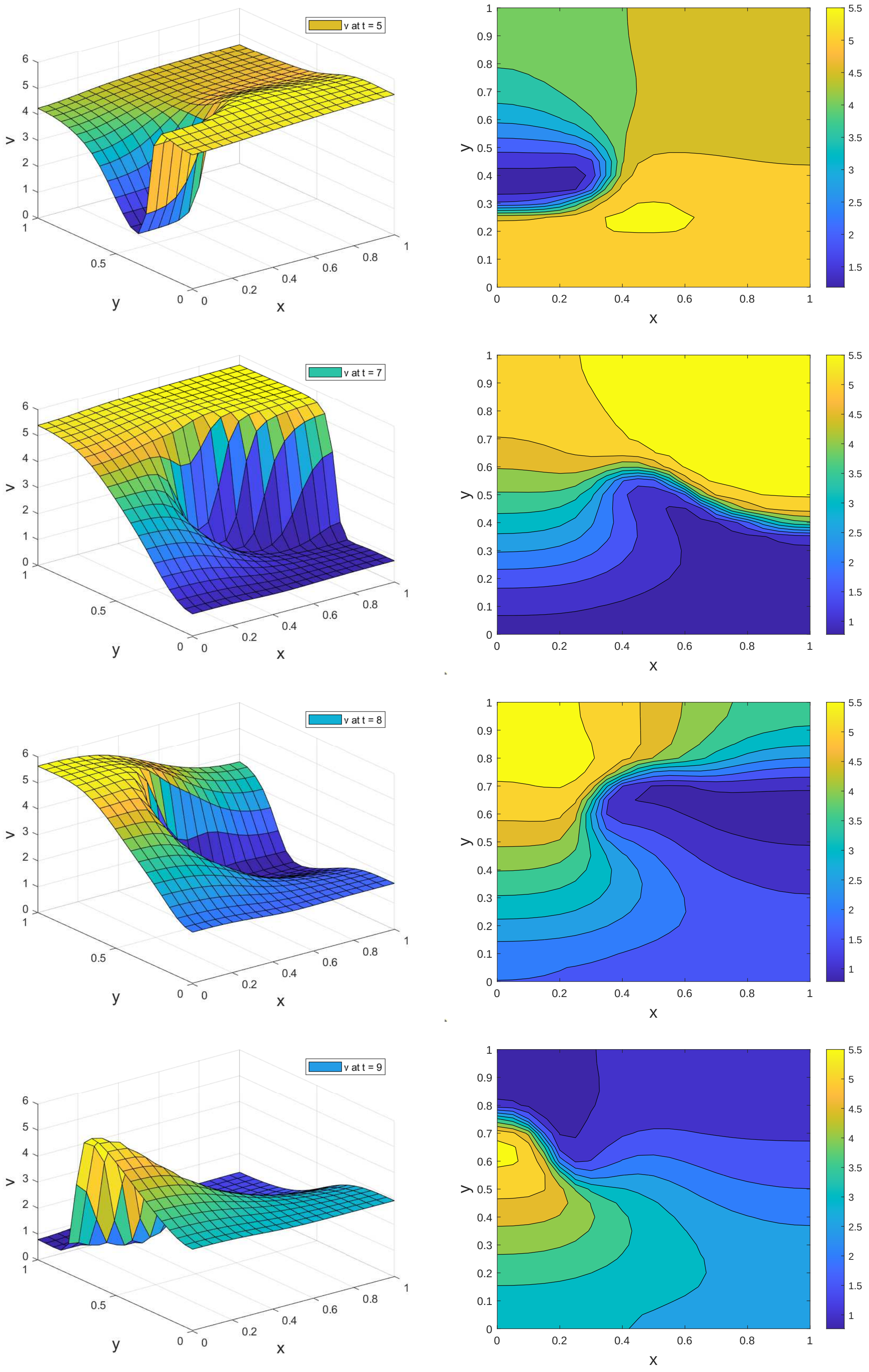


Figure 5.6. The physical behaviour of concentration of v in 3D and its contour of Problem 5.2 at times $t = 5, 7, 8, 9, 10.5, 12.2, 14.2, 15$.

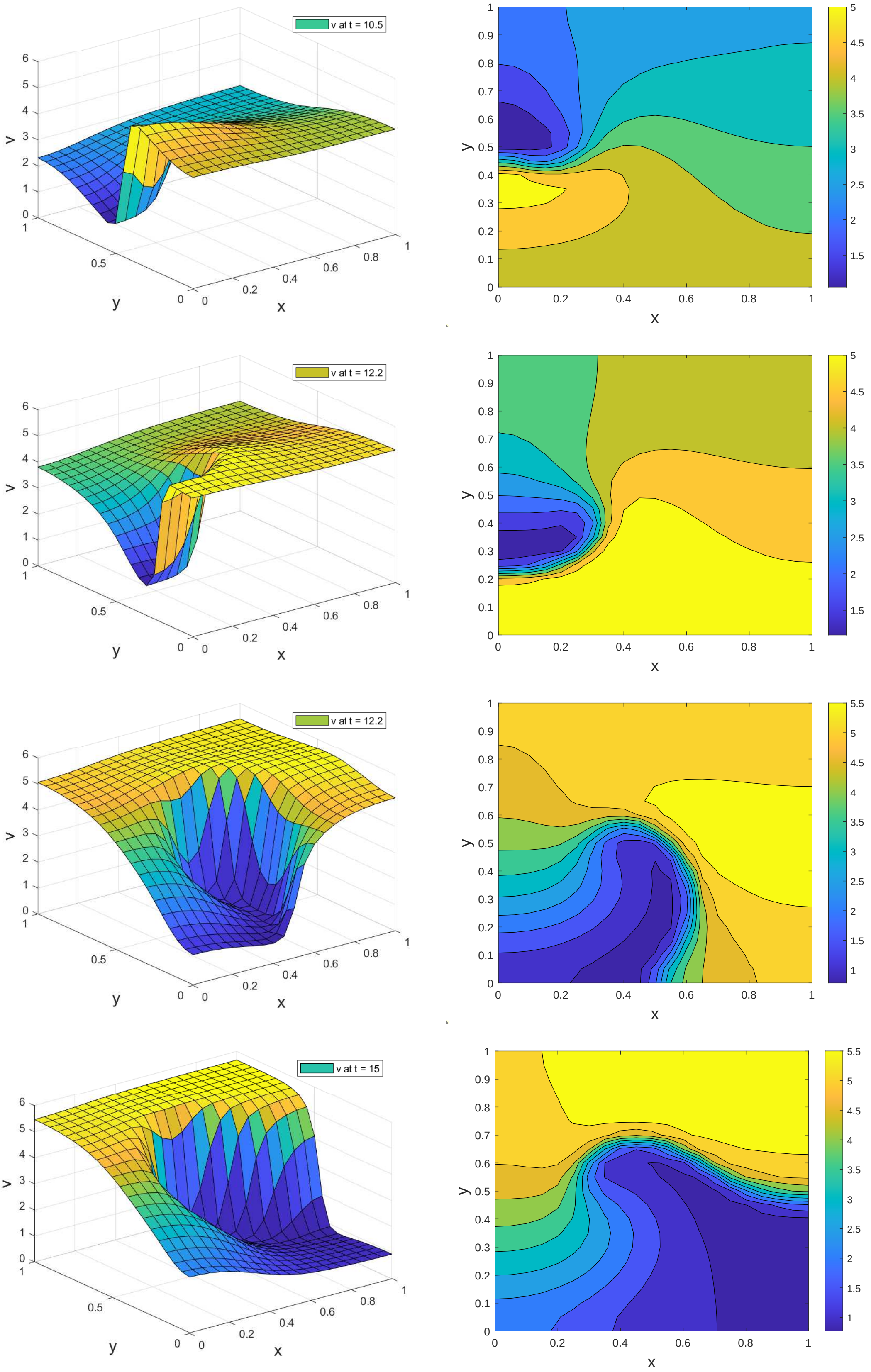


Figure 5.6. continued

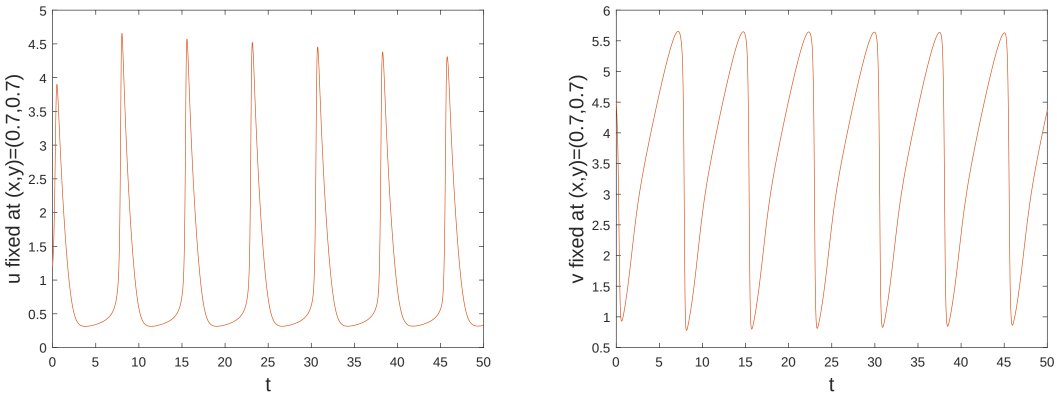


Figure 5.7. The solution profile of u and v of Problem 5.2 fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$.

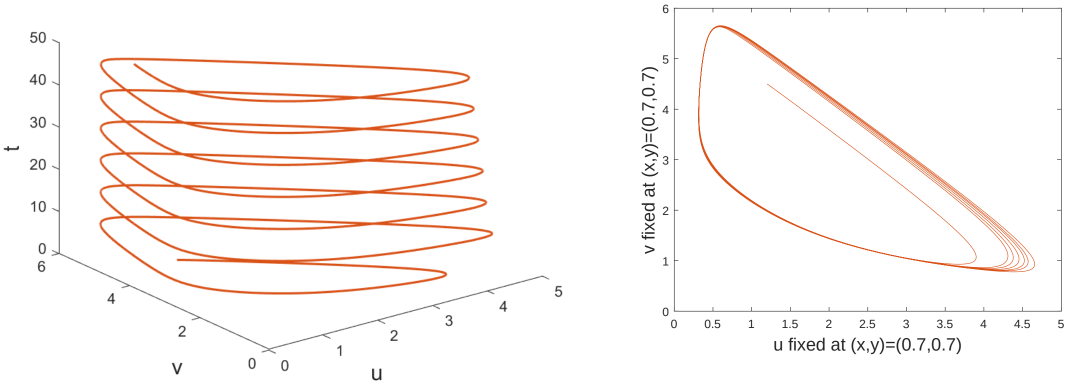


Figure 5.8. The path taken by (u, v) as time increase (left) and the (u, v)-plane (right) of Problem 5.2 fixed at $(x, y) = (0.7, 0.7)$ over $0 < t \leq 50$ for $\hat{\alpha} = 1$, $\hat{\beta} = 3.4$ and $\hat{\gamma} = 0.002$.

Problem 5.3 *Consider the two dimensional Brusselator equation over the domain* $\mathfrak{D} = \{(x, y) \mid 0 \leq x \leq 1,\ 0 \leq y \leq 1\}$ *with the exact solution given as follows (Jiwari and Yuan, 2014; Kumar et al., 2019; Ali et al., 2010)*

$$u(x, y, t) = \exp(-x - y - 0.5t)$$

$$v(x, y, t) = \exp(x + y + 0.5t)$$

where the initial and boundary data are extracted from this solution.

In Problem 5.3 the accuracy of the proposed method is checked and compared with the results in the literature. All the results are obtained using parameters $\hat{\alpha} = 0$, $\hat{\beta} = 1$, $\hat{\gamma} = 0.25$. L_∞ and relative error norms of the problem are presented in Table 5.3 with an accuracy tolerance $\epsilon = 10^{-4}$ in time integration at time $t = 2$ while the comparison of the same error norms with the results obtained in (Kumar et al., 2019) are given in Table 5.4. Table 5.5 lists error norms for different accuracy tolerance. L_∞ error norms is calculated using

$$L_\infty = \max_{1 \leq i \leq N} \sum_{j=1}^{N} |u_{i,j} - \tilde{u}_{i,j}|$$

where $u_{i,j}$ is the numerical solution and $\tilde{u}_{i,j}$ is the exact solution at some later time. The estimated solutions of the problem and results obtained by (Ali et al., 2010) at $(x, y) = (0.4, 0.6)$ at various times with comparison to the exact solutions are shown in Table 5.6. To show the performance of the method the average of Δt's and the CPU times are listed in Table 5.7 for different mesh grids and accuracy tolerances. Some illustrations are provided in Figures 5.9 and 5.10.

Table 5.3. L_∞ and relative error norms at $t = 2$ with the accuracy tolerance $\epsilon = 10^{-4}$.

N	u		v	
	E_∞	E_R	E_∞	E_R
10×10	$6.4521E - 07$	$1.3793E - 05$	$1.1565E - 05$	$7.1135E - 06$
15×15	$9.0706E - 07$	$2.0203E - 06$	$1.2555E - 05$	$7.0631E - 07$
21×21	$1.7645E - 06$	$3.0293E - 05$	$2.0922E - 05$	$7.9491E - 06$

Table 5.4. L_∞ norm errors of Problem 5.3 at $t = 2$ with the accuracy tolerance $\epsilon = 10^{-4}$.

N	(Kumar et al., 2019)		(Kumar et al., 2019)		CFD6	
	u	v	u	v	u	v
10×10	1.6947×10^{-06}	8.5877×10^{-05}	2.6828×10^{-06}	1.3779×10^{-04}	6.4521×10^{-07}	1.1565×10^{-05}
15×15	1.5364×10^{-06}	7.9857×10^{-05}	2.7051×10^{-06}	1.3426×10^{-04}	9.0706×10^{-07}	1.2555×10^{-05}
21×21	1.3452×10^{-06}	1.017×10^{-06}	1.6834×10^{-06}	1.0423×10^{-04}	1.7645×10^{-06}	2.0922×10^{-05}

Table 5.5. Comparison of L_∞ norm errors of Problem 5.3 at $t = 2$ with different accuracy tolerances.

N	$\epsilon = 10^{-4}$		$\epsilon = 10^{-6}$		$\epsilon = 10^{-8}$	
	u	v	u	v	u	v
8×8	5.7835×10^{-07}	2.7724×10^{-05}	6.4836×10^{-07}	3.5185×10^{-05}	1.8381×10^{-06}	9.8128×10^{-05}
16×16	1.3193×10^{-06}	1.6918×10^{-05}	3.0941×10^{-08}	1.6012×10^{-06}	1.0369×10^{-07}	5.5758×10^{-05}
32×32	7.3012×10^{-07}	9.6952×10^{-06}	2.1473×10^{-07}	1.1441×10^{-07}	1.7214×10^{-09}	9.1321×10^{-08}
64×64	8.4838×10^{-08}	2.7154×10^{-07}	1.1684×10^{-08}	3.6713×10^{-08}	1.3844×10^{-10}	7.4640×10^{-09}

Table 5.6. Comparison of numerical results produced by presented method(PM) with the results of (Ali et al., 2010) at the point $(0.40, 0.60)$.

t	u					v				
	$N = 10$		$N = 20$			$N = 10$		$N = 20$		
	PM	Result^S	PM	Result^S	Exact	PM	Result^S	PM	Result^S	Exact
0.30	0.3166	0.3174	0.3166	0.3168	0.3166	3.1582	3.158	3.1582	3.158	3.1582
0.60	0.2725	0.2732	0.2725	0.2724	0.2725	3.6693	3.668	3.6693	3.669	3.6693
0.90	0.2346	0.2351	0.2346	0.2347	0.2346	4.2631	4.262	4.2631	3.669	4.2631
1.20	0.2019	0.2024	0.2019	0.2020	0.2019	4.9530	4.952	4.9530	4.953	4.9530
1.50	0.1738	0.1742	0.1738	0.1739	0.1738	5.7546	5.754	5.7546	5.755	5.7546
1.80	0.1496	0.1499	0.1496	0.1496	0.1496	6.6859	6.685	6.6859	6.686	6.6859

^{S} Results by (Ali et al., 2010)

Table 5.7. The values of u and v with the average of Δt's (Avg(Δt)) and CPU times at $t = 2$ with different accuracy tolerances.

N	$\epsilon = 10^{-4}$, at $(0.25, 0.50)$				$\epsilon = 10^{-6}$, at $(0.50, 0.50)$				$\epsilon = 10^{-8}$, at $(0.50, 0.75)$			
	u	v	Avg(Δt)	CPU(s)	u	v	Avg(Δt)	CPU(s)	u	v	Avg(Δt)	CPU(s)
8×8	0.1738	5.7546	0.01398	0.0072	0.1353	7.3891	0.00881	0.0097	0.1054	9.4877	0.00200	0.0383
16×16	0.1738	5.7546	0.00303	0.0495	0.1353	7.3891	0.00300	0.0473	0.1054	9.4877	0.00139	0.0842
32×32	0.1738	5.7546	0.00073	0.5185	0.1353	7.3891	0.00073	0.5345	0.1054	9.4877	0.00073	0.5249

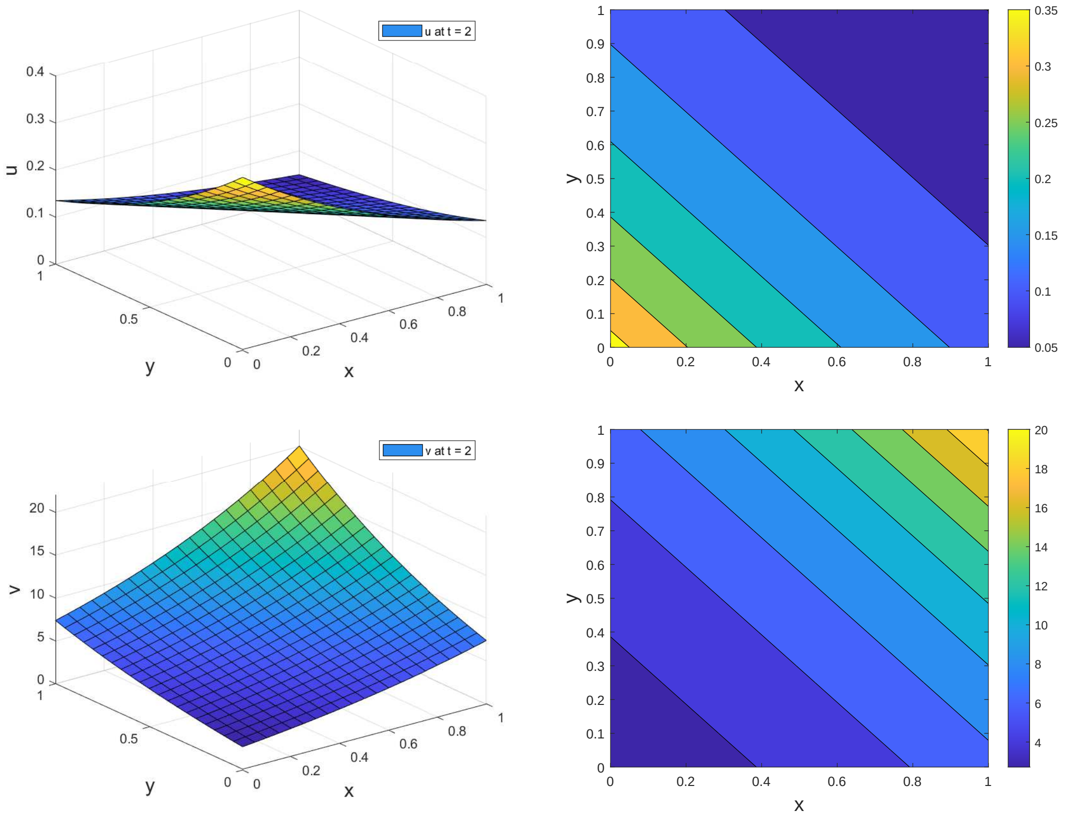


Figure 5.9. The physical behaviour of concentration of u and v in 3D and contour of Problem 5.3 at time $t = 2$.

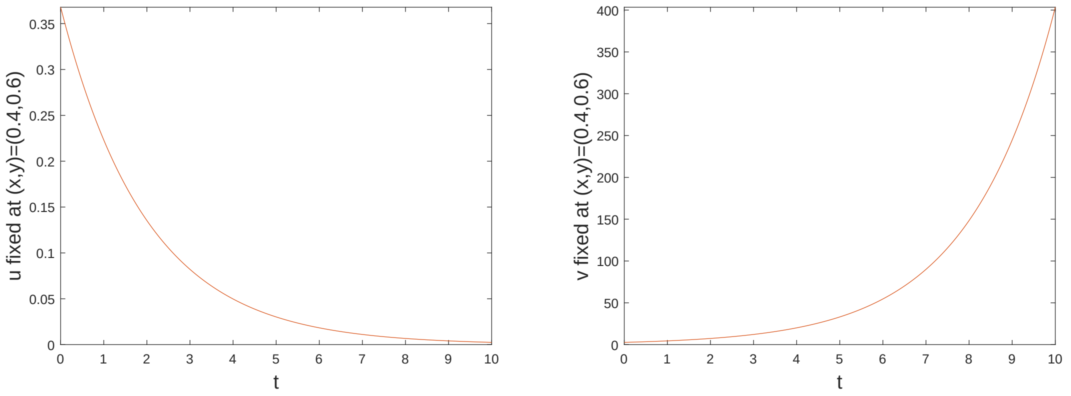


Figure 5.10. The solution profile of u and v of Problem 5.3 fixed at $(x, y) = (0.4, 0.6)$ over $0 < t \leq 10$.

CHAPTER 6

SUMMARY AND CONCLUSION

In this thesis, an elaboration on the construction of Compact Finite Difference formulae is given. To estimate the numerical solution of non-periodic boundary problems one-sided and central difference based compact schemes are combined to form a matrix to approximate the spatial differentials in the model. The method of lines is utilized then to reduce the PDE into a time-dependent ODE which then is integrated by a technique with the step-size control called adaptive step-size Runge-Kutta method. This particular adaptive step-size technique is based on embedded Runge-Kutta formulas, i.e., two formulas with identical evaluation points is used to minimize the computational effort. The investigation of a sixth-order tridiagonal compact scheme for space combined with a fifth-order time integrator is carried throughout the work. The constructed method is applied to Brusselator systems of one and two dimensions. A simple stability analysis of non-linear system of ODEs based on the eigenvalues is used to show the stability of our method when applied to the model. The illustrations and results of three different problems with different parameters are obtained and reveal that the proposed method is capable of producing highly accurate results with minimal computational cost. The results obtained are compared with the best results obtained in (Kumar et al., 2019; Ali et al., 2010). The similarity and accuracy of the results guarantee to conclude that the presented method is an efficient and a reliable alternative for solving Brusselator system.

REFERENCES

Ali, A., S. Haq, et al. (2010). A computational modeling of the behavior of the two-dimensional reaction–diffusion brusselator system. *Applied Mathematical Modelling 34*(12), 3896–3909.

Alqahtani, A. M. (2018). Numerical simulation to study the pattern formation of reaction–diffusion brusselator model arising in triple collision and enzymatic. *Journal of Mathematical Chemistry 56*(6), 1543–1566.

Ang, W.-T. (2003). The two-dimensional reaction–diffusion brusselator system: a dual-reciprocity boundary element solution. *Engineering Analysis with Boundary Elements 27*(9), 897–903.

Bahar, E. and G. Gurarslan (2020). B-spline method of lines for simulation of contaminant transport in groundwater. *Water 12*(6), 1607.

Browder, A., W. Wolfgang, W. Walter, and W. L. Walter (1998). *Ordinary differential equations*, Volume 182. Springer Science & Business Media.

Cicek, Y., N. Gucuyenen Kaymak, E. Bahar, G. Gurarslan, and G. Tangolu (2022). A new numerical algorithm based on quintic b-spline and adaptive time integrator for coupled burger's equation. *Computational Methods for Differential Equations*.

Dormand, J. R. and P. J. Prince (1980). A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics 6*(1), 19–26.

Fehlberg, E. (1968). *Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control*. National Aeronautics and Space Administration.

Fehlberg, E. (1969). *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*, Volume 315. National aeronautics and space administration.

Gurarslan, G., H. Karahan, D. Alkaya, M. Sari, and M. Yasar (2013). Numerical solution of advection-diffusion equation using a sixth-order compact finite difference method. *Mathematical Problems in Engineering 2013*.

Haq, S., I. Ali, and K. S. Nisar (2021). A computational study of two-dimensional reaction–diffusion brusselator system with applications in chemical processes. *Alexandria Engineering Journal 60*(5), 4381–4392.

Hoffmann, K. A. and S. T. Chiang (2000). Computational fluid dynamics volume i. *Engineering education system*.

İmamoğlu Karabaş, N., S. Ö. Korkut, G. Gurarslan, and G. Tanoğlu (2022). A reliable and fast mesh-free solver for the telegraph equation. *Computational and Applied Mathematics 41*(5), 1–24.

Jena, R. M., S. Chakraverty, H. Rezazadeh, and D. Domiri Ganji (2020). On the solution of time-fractional dynamical model of brusselator reaction-diffusion system arising in chemical reactions. *Mathematical Methods in the Applied Sciences 43*(7), 3903–3913.

Jiwari, R. and J. Yuan (2014). A computational modeling of two dimensional reaction–diffusion brusselator system arising in chemical processes. *Journal of mathematical Chemistry 52*(6), 1535–1551.

Kumar, S., R. Jiwari, and R. Mittal (2019). Numerical simulation for computational modelling of reaction–diffusion brusselator model arising in chemical processes. *Journal of Mathematical Chemistry 57*(1), 149–179.

Lefever, R. and G. Nicolis (1971). Chemical instabilities and sustained oscillations. *Journal of theoretical Biology 30*(2), 267–284.

Lele, S. K. (1992). Compact finite difference schemes with spectral-like resolution. *Journal of computational physics 103*(1), 16–42.

Mittal, R. and R. Jiwari (2011). Numerical solution of two-dimensional reaction–diffusion brusselator system. *Applied mathematics and computation 217*(12), 5404–5415.

Mittal, R., S. Kumar, and R. Jiwari (2022). A cubic b-spline quasi-interpolation algorithm to capture the pattern formation of coupled reaction-diffusion models. *Engineering with Computers 38*(2), 1375–1391.

Mittal, R. and R. Rohila (2016). Numerical simulation of reaction-diffusion systems by modified cubic b-spline differential quadrature method. *Chaos, Solitons & Fractals 92*, 9–19.

Onarcan, A. T., N. Adar, and I. Dag (2018). Trigonometric cubic b-spline collocation algorithm for numerical solutions of reaction–diffusion equation systems. *Computational and Applied Mathematics 37*(5), 6848–6869.

Press, W. H. and S. A. Teukolsky (1992). Adaptive stepsize runge-kutta integration. *Computers in Physics 6*(2), 188–191.

Prigogine, I. and R. Lefever (1968). Symmetry breaking instabilities in dissipative systems. ii. *The Journal of Chemical Physics 48*(4), 1695–1700.

Prigogine, I. and G. Nicolis (1985). Self-organisation in nonequilibrium systems: towards a dynamics of complexity. In *Bifurcation analysis*, pp. 3–12. Springer.

Prince, P. J. and J. R. Dormand (1981). High order embedded runge-kutta formulae. *Journal of computational and applied mathematics 7*(1), 67–75.

Saad, K. M. (2021). Fractal-fractional brusselator chemical reaction. *Chaos, Solitons & Fractals 150*, 111087.

Sari, M. and G. Gürarslan (2009). A sixth-order compact finite difference scheme to the numerical solutions of burgers' equation. *Applied Mathematics and Computation 208*(2), 475–483.

Sari, M., G. Gürarslan, and İ. Dağ (2010). A compact finite difference method for the solution of the generalized burgers–fisher equation. *Numerical Methods for Partial Differential Equations: An International Journal 26*(1), 125–134.

Twizell, E., A. Gumel, and Q. Cao (1999). A second-order scheme for the ”brusselator” reaction–diffusion system. *Journal of Mathematical Chemistry 26*(4), 297–316.

Wazwaz, A.-M. (2000). The decomposition method applied to systems of partial differential equations and to the reaction–diffusion brusselator model. *Applied mathematics and computation 110*(2-3), 251–264.

APPENDIX A

MATLAB CODES FOR NUMERICAL RESULTS

A.1. Codes for Problem 5.1

```
clear;clc;
%% Time interval, space interval
global nx dx D_M2
xmin = 0; xmax = 1;
tmin = 0; tmax = 100;
nx = 21;
dx = (xmax-xmin)/(nx-1);
x = (xmin:dx:xmax)';
%% initial condition
f1 = @(x) 0.5 + 0*x;
f2 = @(x) 1 + 5*x;
U0 = f1(x);  V0 = f2(x);
u = U0;      v = V0;
a = 11;
u0a = u(a); v0a = v(a);
tic
D_M2 = CFD6M2(dx, nx);
y=vertcat(u,v);
t=tmin; T = 0;
tol=1e-4;
h=tol^(1/5)/4;
step=0; fcall=1; nrej = 0;
a4=[44/45 -56/15 32/9]';
a5=[19372/6561 -25360/2187 64448/6561 -212/729]';
```

```
a6=[9017/3168 -355/33 46732/5247 49/176 -5103/18656]’;
a7=[35/384 0 500/1113 125/192 -2187/6784 11/84]’;
e=[71/57600 -1/40 -71/16695 71/1920 -17253/339200 22/525]’;
k1=func(t,y);
while t < tmax
y=vertcat(u,v);
if t+h > tmax; h=tmax-t; end
k2=func(t+h/5,y+h*k1/5);
k3=func(t+3*h/10,y+h*(3*k1+9*k2)/40);
k4=func(t+4*h/5,y+h*(a4(1)*k1+a4(2)*k2+a4(3)*k3));
k5=func(t+8*h/9,y+h*(a5(1)*k1+a5(2)*k2+a5(3)*k3+...
a5(4)*k4));
k6=func(t+h,y+h*(a6(1)*k1+a6(2)*k2+a6(3)*k3+a6(4)*k4+...
a6(5)*k5));
yt=y+h*(a7(1)*k1+a7(3)*k3+a7(4)*k4+a7(5)*k5+a7(6)*k6);
k2=func(t+h,yt);
est=norm(h*(e(1)*k1+e(2)*k2+e(3)*k3+e(4)*k4+e(5)*k5+...
e(6)*k6),inf);
fcall=fcall+6;

if est < tol
t=t+h;
k1=k2;
step=step+1;
u = reshape(yt(1:nx),nx,1);
v = reshape(yt(nx+1:end),nx,1);
u(1) = u(2); u(end,:) = u(end-1);
v(1) = v(2); v(end,:) = v(end-1);
T = [T t];
u0a = [u0a u(a)];
v0a = [v0a v(a)];
else
```

```
nrej=nrej+1;
end
h=.9*min((tol/(est+eps))^(1/5),10)*h;
end
toc
figure(6)
plot3(u0a,v0a,T, 'Color','#D95319','Linewidth', 1.5)
xlabel('u', 'FontSize',15)
ylabel('v', 'FontSize',15)
zlabel('t', 'FontSize',15)
figure(5)
plot(u0a, v0a, 'Color','#D95319');
axis ([0 3.5 0 4])
xlabel('u fixed at x = 0.5', 'FontSize',15)
ylabel('v fixed at x = 0.5', 'FontSize',15)
% figure(1)
% plot(T, u0a, 'Color','#D95319')
% xlabel('t', 'FontSize', 15)
% ylabel('u fixed at x=0.5', 'FontSize', 15)
% axis([tmin tmax 0.2 3.5])
% figure(2)
% plot(T, v0a,'Color', '#D95319')
% xlabel('t', 'FontSize',15)
% ylabel('v fixed at x=0.5', 'FontSize',15)
% axis([tmin tmax 0 4])
% figure(1)
% plot(x, u)
% xlabel('x', 'FontSize',15)
% ylabel('u and v', 'FontSize',15)
% hold on
% plot(x, v)
% legend('u at t=15','v at t=15')
% hold off
% axis([xmin xmax 0 6])

function k = func(t, u)
global nx dx D_M2
%% constant
k1 = 0.0001; alpha = 1; beta = 0.5;

k = [k1*D_M2*u(1:nx) + alpha - (beta+1)*u(1:nx) + ...
    (u(1:nx).^2).*u(nx+1:end);k1*D_M2*u(nx+1:end) + ...
    beta*u(1:nx) - (u(1:nx).^2).*u(nx+1:end) + 0*t];
end
```

A.2. Codes for Problem 5.2

```
clear;clc;
%% Time interval, space interval
xmin = 0; xmax = 1; ymin = 0; ymax = 1; tmin = 0; tmax = 5;
global nx ny dx dy D_M2
nx = 21; ny = 21; dx=(xmax-xmin)/(nx-1);
dy=(ymax-ymin)/(ny-1); dt = 0.01;
x = (xmin:dx:xmax)'; y = (ymin:dx:ymax)';
[X, Y] = meshgrid(x,y);

%% initial condition
f1 = @(x,y) 0.5 + y + 0*x;
f2 = @(x,y) 1 + 5*x + 0*y;

U0 = f1(X,Y);  V0 = f2(X,Y);
U = U0;        V = V0;

tic
```

```
D_M2 = CFD6M2(dx, nx);
t = tmin; T = tmin;
step = 0; nrej = 0;
% a = 15;
% U0a = U(a,a); V0a = V(a,a);
eps=1e-4;
while t < tmax
    if t+dt > tmax; dt=tmax-t; end
     [F1, G1] = FUN(U, V);
    [F2, G2] = FUN(U+dt*F1, V+dt*G1);
    [F3, G3] = FUN(U+(dt/4)*(F1+F2), V+(dt/4)*(G1+G2));

    e = (dt/3)*max(norm(F1-2*F3+F2), norm(G1-2*G3+G2));
    if e <= eps
        U = U + (1/6)*dt*(F1+4*F3+F2);v
        V = V + (1/6)*dt*(G1+4*G3+G2);
        t = t + dt;
        step = step + 1;
        U(1,:) = U(2,:); U(end,:) = U(end-1,:);
        U(:,1) = U(:,2); U(:,end) = U(:,end-1);
        V(1,:) = V(2,:); V(end,:) = V(end-1,:);
        V(:,1) = V(:,2); V(:,end) = V(:,end-1);

%         U0a = [U0a U(a,a)];
%         V0a = [V0a V(a,a)];
%         T = [T t];
    else
        nrej = nrej + 1;
    end
        dt=0.9*dt*(eps/e)^(1/3);
end
toc
```

```
% figure(7)
% surf(X, Y, U’); axis([xmin xmax ymin ymax 0 6])
% xlabel(’x’, ’FontSize’,15)
% ylabel(’y’, ’FontSize’,15)
% zlabel(’v’, ’FontSize’,15)
% figure(8)
% contourf(X, Y, V’)
% xlabel(’x’, ’FontSize’,15)
% ylabel(’y’, ’FontSize’,15)
% colorbar
figure(1)
surf(X, Y, V’); axis([xmin xmax ymin ymax 0 6])
xlabel(’x’, ’FontSize’,15)
ylabel(’y’, ’FontSize’,15)
zlabel(’v’, ’FontSize’,15)
legend(’v at t = 15’,’Location’,’northeast’)
% colorbar
figure(2)
contourf(X, Y, V’)
xlabel(’x’, ’FontSize’,15)
ylabel(’y’, ’FontSize’,15)
colorbar
% figure(6)
% plot3(U0a,V0a,T, ’Color’,’#D95319’,’Linewidth’, 1.5)
% xlabel(’u’, ’FontSize’,15)
% ylabel(’v’, ’FontSize’,15)
% zlabel(’t’, ’FontSize’,15)
% figure(5)
% plot(U0a, V0a, ’Color’,’#D95319’);
% axis ([0 5 0 6])
% xlabel(’u fixed at (x,y)=(0.7,0.7)’, ’FontSize’,15)
% ylabel(’v fixed at (x,y)=(0.7,0.7)’, ’FontSize’,15)
% figure(3)
% plot(T, U0a, 'Color','#D95319');
% axis([tmin tmax 0.5 5])
% xlabel('t', 'FontSize',15)
% ylabel('u fixed at (x,y)=(0.7,0.7)', 'FontSize',15)
% figure(4)
% plot(T, V0a, 'Color','#D95319');
% axis([tmin tmax 0 5])
% xlabel('t', 'FontSize',15)
% ylabel('v fixed at (x,y)=(0.7,0.7)', 'FontSize',15)

function [F, G] = FUN(U, V)
global D_M2
%% constant
k = 0.002; alpha = 1; beta = 3.4;
Uxx = D_M2*U'; Uyy = D_M2*U;
Vxx = D_M2*V'; Vyy = D_M2*V;
F = k*(Uxx' + Uyy) + alpha - (beta+1)*U + (U.^2).*V;
G = k*(Vxx' + Vyy) + beta*U - (U.^2).*V;
end
```

A.3. Codes for Problem 5.3

```
clear;clc;
%% Time interval, space interval
xmin = 0; xmax = 1; ymin = 0; ymax = 1; tmin = 0; tmax = 2;
global nx ny nxy dx dy D_M2
nx=11; ny = 11;
dx=(xmax-xmin)/(nx-1); dy=(ymax-ymin)/(ny-1);
xx = (xmin:dx:xmax); yy = (ymin:dx:ymax);
[X, Y] = meshgrid(xx,yy);
nxy = nx*ny;
```

```
%% exact solution
exct_u=@(x,y,t) exp(-x-y-0.5*t);
exct_v=@(x,y,t) exp(x+y+0.5*t);
u0 = exct_u(X,Y,tmin);       v0 = exct_v(X,Y,tmin);
u = reshape(u0,nxy,1); v = reshape(v0,nxy,1);
tic
D_M2 = CFD6M2(dx, nx);
y=vertcat(u,v);
t=tmin; T = 0;
a = 5; b = 7;
u0a = u0(a,b); v0a = v0(a,b);
tol=1e-4;
h=tol^(1/5)/4;
step=0; fcall=1; nrej = 0;
a4=[44/45 -56/15 32/9]’;
a5=[19372/6561 -25360/2187 64448/6561 -212/729]’;
a6=[9017/3168 -355/33 46732/5247 49/176 -5103/18656]’;
a7=[35/384 0 500/1113 125/192 -2187/6784 11/84]’;
e=[71/57600 -1/40 -71/16695 71/1920 -17253/339200 22/525]’;
k1=func(t,y);

while t < tmax
y=vertcat(u,v);
if t+h > tmax; h=tmax-t; end
k2=func(t+h/5,y+h*k1/5);
k3=func(t+3*h/10,y+h*(3*k1+9*k2)/40);
k4=func(t+4*h/5,y+h*(a4(1)*k1+a4(2)*k2+a4(3)*k3));
k5=func(t+8*h/9,y+h*(a5(1)*k1+a5(2)*k2+a5(3)*k3+...
a5(4)*k4));
k6=func(t+h,y+h*(a6(1)*k1+a6(2)*k2+a6(3)*k3+a6(4)*k4+...
a6(5)*k5));
yt=y+h*(a7(1)*k1+a7(3)*k3+a7(4)*k4+a7(5)*k5+a7(6)*k6);
```

```
k2=func(t+h,yt);
est=norm(h*(e(1)*k1+e(2)*k2+e(3)*k3+e(4)*k4+e(5)*k5+...
e(6)*k6),inf);
fcall=fcall+6;

if est < tol
t=t+h;
k1=k2;
step=step+1;
ut = reshape(yt(1:nxy),ny,nx);
vt = reshape(yt(nxy+1:end),ny,nx);
ut(1,:) = exct_u(xmin,yy,t);
vt(1,:) = exct_v(xmin,yy,t);
ut(:,1) = exct_u(xx,ymin,t);
vt(:,1) = exct_v(xx,ymin,t);
ut(end,:) = exct_u(xmax,yy,t);
vt(end,:) = exct_v(xmax,yy,t);
ut(:,end) = exct_u(xx,ymax,t);
vt(:,end) = exct_v(xx,xmax,t);
T = [T t];
u0a = [u0a ut(a,b)];
v0a = [v0a vt(a,b)];
u = reshape(ut,nxy,1);
v = reshape(vt,nxy,1);
else
nrej=nrej+1;
end
h = .9*min((tol/(est+eps))^(1/5),10)*h;
end
toc
error_u = norm(ut - exct_u(X,Y,tmax),inf);
error_v = norm(vt - exct_v(X,Y,tmax),inf);
```

```
figure(1)
surf(X, Y, vt); axis([xmin xmax ymin ymax 0 22])
xlabel(’x’, ’FontSize’,15)
ylabel(’y’, ’FontSize’,15)
zlabel(’v’, ’FontSize’,15)
legend(’v at t = 2’,’Location’,’northeast’)
figure(2)
contourf(X,Y,vt)
xlabel(’x’, ’FontSize’,15)
ylabel(’y’, ’FontSize’,15)
colorbar
figure(3)
surf(X, Y, ut); axis([xmin xmax ymin ymax 0 0.4])
xlabel(’x’, ’FontSize’,15)
ylabel(’y’, ’FontSize’,15)
zlabel(’u’, ’FontSize’,15)
legend(’u at t = 2’,’Location’,’northeast’)
figure(4)
contourf(X,Y,ut)
xlabel(’x’, ’FontSize’,15)
ylabel(’y’, ’FontSize’,15)
colorbar

function k = func(t, y)
global nx ny nxy D_M2
%% constant
k1 = 0.25; alpha = 0; beta = 1;
Ut = reshape(y(1:nxy),ny,nx);
Vt = reshape(y(nxy+1:end),ny,nx);
Uxx = D_M2*Ut’; Uyy = D_M2*Ut;
Vxx = D_M2*Vt’; Vyy = D_M2*Vt;
```

```
Uk = k1*(Uxx' + Uyy) + alpha - (beta+1)*Ut + (Ut.^2).*Vt;
Vk = k1*(Vxx' + Vyy) + beta*Ut - (Ut.^2).*Vt + 0*t;
U = reshape(Uk,nxy,1); V = reshape(Vk,nxy,1);
k = vertcat(U,V);
end
```

A.4. Approximation Matrix for Second-order Derivative

```
function M2 = CFD6M2(dx, nsize)
A = zeros(nsize);
B = zeros(nsize);
i = 1;
A(i,i:i+1) = [1 11];
B(i,i:i+3) = [13 -27 15 -1];
i = 2;
A(i,i-1:i+1) = [1/10 1 1/10];
B(i,i-1:i+1) = [6/5 -12/5 6/5];
for i = 3:nsize-2
    A(i,i-1:i+1) = [2/11 1 2/11];
    B(i,i-2:i+2) = [3/44 12/11 -51/22 12/11 3/44];
end
i = nsize-1;
A(i,i-1:i+1) = [1/10 1 1/10];
B(i,i-1:i+1) = [6/5 -12/5 6/5];
i = nsize;
A(i,i-1:i) = [11 1];
B(i,i-3:i) = [-1 15 -27 13];
B = (1/dx^2)*B;
M2 = A\B;
end
```