NUMERICAL SOLUTION OF SEMI-LINEAR ADVECTION-DIFFUSION-REACTION EQUATIONS BY DISCONTINUOUS GALERKIN METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SÜLEYMAN YILDIZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

JUNE 2016

Approval of the thesis:

NUMERICAL SOLUTION OF SEMI-LINEAR ADVECTION-DIFFUSION-REACTION EQUATIONS BY DISCONTINUOUS GALERKIN METHODS

submitted by **SÜLEYMAN YILDIZ** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _____

Assoc. Prof. Dr. Ömür Uğur
Head of Department, **Scientific Computing** _____

Prof. Dr. Bülent Karasözen
Supervisor, **Scientific Computing, METU** _____

Examining Committee Members:

Prof. Dr. Bülent Karasözen
Department of Mathematics & Institute of Applied Mathematics, METU _____

Prof. Dr. Gerhard Wilhelm Weber
Institute of Applied Mathematics, METU _____

Assoc. Prof. Dr. Ayhan Aydın
Department of Mathematics, Atılım University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: SÜLEYMAN YILDIZ

Signature :

ABSTRACT

NUMERICAL SOLUTION OF SEMI-LINEAR
ADVECTION-DIFFUSION-REACTION EQUATIONS BY DISCONTINUOUS
GALERKIN METHODS

YILDIZ, SÜLEYMAN
M.S., Department of Scientific Computing
Supervisor : Prof. Dr. Bülent Karasözen

JUNE 2016, 41 pages

In this thesis, we study splitting methods for semi-linear advection-diffusion-reaction (ADR) equations which are discretized by the symmetric interior penalty Galerkin (SIPG) method in space. For the time integration Rosenbrock methods are used with Strang splitting. The linear system of equations are solved iteratively by preconditioned generalized minimum residual method (GMRES). Numerical experiments for ADR equations with different type nonlinearities demonstrate the effectiveness of the proposed approach.

Keywords : the discontinuous Galerkin (dG) method, operator splitting, Strang splitting, Rosenbrock methods, advection-diffusion-reaction equation

ÖZ

YARI-DOĞRUSAL ADVEKSİYON-DİFÜZYON-REAKSİYON DENKLEMLERİNİN SÜREKSİZ GALERKİN YÖNTEMİYLE NÜMERİK ÇÖZÜMLERİ

YILDIZ, SÜLEYMAN
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Bülent Karasözen

Haziran 2016, 41 sayfa

Bu tezde, zaman ayırma metodları uzayda simetrik süreksiz Galerkin yöntemi ile yarı-doğrusal adveksiyon-difüzyon-reaksiyon (ADR) denklemleri için incelenmiştir. Denklemin zaman integrallemesi için Rosenbrock metodları ve Strang operatör ayırması kullanılmıştır. Lineer sistem preconditioner kullanılarak generalized minimum residual method (GMRES) ile iteratif bir biçimde çözülmüştür. Nümerik çözümlerin doğrusal olmayan farklı ADR örnekleri için verimliliği sunulmuştur.

Anahtar Kelimeler : Strang ayırması, süreksiz Galerkin yöntemleri, Rosenbrock metodları, adveksiyon-reaksiyon-difüzyon denklemleri

To My Family

ACKNOWLEDGMENTS

I would like to express my very great appreciation to my thesis supervisor Prof. Dr. Bülent Karasözen for his patient guidance, enthusiastic encouragement and valuable advices during the development and preparation of this thesis. His willingness to give his time and to share his experiences has brightened my path.

I would like to thank to Dr. Hamdullah Yücel for his useful comments and clarifying discussions on discontinuous Galerkin methods.

I also would like to thank Bilgi Yılmaz, Güray Kara, Abdullah Ali Sivas and all other friends for their useful comments.

Lastly, special thanks to my beloved wife and to my family for their patience and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Many real-life applications such as physical, biological, chemical and financial modelling are fundamental processes of reaction, convection, diffusion phenomenons. In partial differential equations (PDEs), convection appears in form of transport mechanism of a substance or conserved property and contribution of diffusion to the system is movement of a substance from high concentration to low concentration, reaction refers response of the mechanism.

Finite element methods (FEMs) have been accepted as an accurate and efficient method for solving PDEs. An advantage of FEMs is their ability to handle complicated geometries. Another advantage of using the FEMs is their ability to use higher order approximations. On the other hand, the disadvantage of FEMs is that they do not have local mass conservation property. Finite volume method (FVM) can be a proper choice for local mass conservation property, however FVM has a lack of ability to use higher order approximations. Combining the finest features of the FVM and FEMs, discontinuous Galerkin (dG) method is an attractive and accurate method in flow and transport problems.

In this thesis, we consider the semi-linear advection-diffusion reaction (ADR) equations of the form

$$\begin{aligned} \frac{\partial u}{\partial t} - \epsilon \Delta u + \vec{b}\,(x,t)\,\nabla u + r(u) &= f\,(x,t) && \text{in} \quad \Omega \times (0,T]\,, \\ u(x,t) &= g^D && \text{on} \quad \Gamma_D \times (0,T]\,, \\ \epsilon \nabla u(x,t) \cdot \vec{n} &= g^N && \text{on} \quad \Gamma_N \times (0,T]\,, \\ u(x,0) &= u_0 && \text{in} \quad \Omega, \end{aligned} \tag{1.1}$$

with Ω is bounded, open, convex domain in $\mathbb{R}^2$ with boundaries $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Here, ϵ is the diffusivity constant, $f\,(x,t) \in L^2(\Omega)$ is the source function, $\vec{b}\,(x,t) \in (W^{1,\infty}(\Omega))^2$ is the velocity field, $g^D \in H^{3/2}(\Gamma_D)$ is the Dirichlet boundary condition, $g^N \in H^{1/2}(\Gamma_N)$ is the Neumann boundary condition, $u_0 \in L^2(\Omega)$ is the initial condition and $\vec{n}$ denote outward normal vector to the boundary.

The aim of this thesis is to investigate the accuracy and efficiency of Rosenbrock time

integrators and the Strang time splitting for semi-linear ADR equations using dG discretization in space. For stiff ordinary differential equations (ODEs) or differential algebraic equations(DAEs), Rosenbrock methods offers many advantages like ease of implementation and less computational complexity. In recent years, numerous works have been conducted on the development of Rosenbrock methods [11, 21, 24, 23], which are applied to various fields such as biogeochemical processes [32], photochemical dispersion problems [38] and electric circuit simulation [17].

Throughout the last century, with new physical phenomenas, the complexity of the equations has increased because of the increase in the complexity of the phenomenas which the equations are involved. In order to solve these problems, new solving methods have been developed. Thus, the operator splitting methods are attractive solvers since they divide the problem into simpler subproblems. Yet, operator splitting methods are not only used for simplifying the complexity, but also with different combination of splitted parts higher order methods can be obtained. Most popular of operator splitting method proposed by Strang [34], called Strang splitting.

A weakness with dG methods is that the linear systems derived from dG methods are large and generally ill-conditioned. In order to deal with this large linear systems the generalized minimum residual method (GMRES) [31] is a suitable choice. Efficiency of the GMRES method can be increase by preconditioners so that selection of preconditioner is an important topic. Several preconditioning are designed for linear systems arising from discontinuous Galerkin discretization [28, 1] in recent years. We use here the GMRES solver with the preconditioner designed for non-symmetric linear systems arising from dG discretization of ADR equations in [15]

The outline of this thesis is as follows: In Chapter 2, we give brief overview about the interior penalty discontinuous Galerkin (IPG) methods. We give the semi-discrete IPG formulation for Equation (1.1) with upwinding for convection. In Chapter 3, we introduce the Rosenbrock methods. Then, we give the fully discrete formulation of Equation (1.1) by using symmetric interior penalty Galerkin (SIPG) method in space discretization and Rosenbrock methods in time discretization. At the end of the Chapter 3, we introduce operator splitting methods and Strang splitting method. Then, we combine a second order 2-stage Rosenbrock integrator ROS2 method and explicit trapezoid rule with Strang splitting method [38] for Equation (1.1). The Rosenbrock solver ROS2 is an second order L-stable method [38]. On the other hand, it is pointed out that the trapezoidal rule is one of the most accurate A-stable method [33]. Hence, by combining Strang splitting with explicit trapezoid rule and ROS2 Method, we investigate the efficiency and accuracy of this combination. In Chapter 4, we present numerical solution of three advection-diffusion-reaction problems with the methods which are presented in Chapter 3. We present the convergence rate of the problems by $L^2(L^2)$ and $L^2(H^1)$ norm. Efficient solution of linear system of equations arising from dG discretization by the preconditioned GMRES method is presented in Chapter 4. The thesis ends with some conclusions in Chapter 5.

CHAPTER 2

DISCONTINUOUS GALERKIN SPATIAL DISCRETIZATION

In the early 1970s, the first discontinuous Galerkin (dG) was proposed and analyzed by Reed and Hill [29] as an alternative to high-order finite difference and finite volume methods. Later on the dG methods become popular for solving hyperbolic problems. In the late 1970s, Douglas and Dupont have first introduced interior penalty (IP) methods [13] for parabolic and elliptic problems. Then, in the eighties, several studies had been done on elliptic problems [4, 8] and for problems with advection in [3, 5, 20]. The well-known dG methods are the local discontinuous Galerkin (LDG) method for diffusion-convection equations proposed by Cockburn and Shu [10], and the compact discontinuous Galerkin (CDG) method proposed by Peraire and Persson [27].

In this Chapter we show detailed construction of the symmetric interior penalty Galerkin (SIPG) method for semi-linear ADR equations following [2, 30].

2.1 Preliminaries

In this Section, we introduce some useful definitions which are required in the construction of discontinuous interior point Galerkin methods.

2.1.1 Sobolev spaces

The spaces $L^p(\Omega)$ of p-integrable functions are defined by

$$L^p(\Omega) = \left\{ v \text{ Lebesgue measurable} : \|v\|^2_{L^p(\Omega)} < \infty \right\}, \qquad 1 \leq p \leq \infty,$$

where Ω polygonal domain in $\mathbb{R}^d$. And the associated norm is defined by

$$\|v\|^2_{L^p(\Omega)} = \left(\int_\Omega |v(x)|^p dx \right)^{\frac{1}{p}}.$$

We mainly consider the space $L^2(\Omega)$ which is a Hilbert space equipped with the usual L^2-inner product

$$(u, v)_\Omega = \int_\Omega u(x)v(x)dx, \qquad \|v\|^2_{L^2(\Omega)} = \sqrt{(v, v)_\Omega}.$$

Let $\mathcal{D}(\Omega)$ denotes the subspace of the space C^∞ having compact support in Ω. For any multi-index $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ with $|\alpha| = \sum_{i=1}^d \alpha_i$, the distributional derivative $D^\alpha v$ is defined by

$$D^\alpha v(\psi) = (-1)^{|\alpha|} \int_\Omega v(x) \frac{\partial^{|\alpha|}\psi}{\partial^{\alpha_1} x_1 \cdots \partial^{\alpha_d} x_d}, \qquad \forall \psi \in \mathcal{D}(\Omega).$$

Then, we define for an integer s the Sobolev spaces

$$H^s(\Omega) = \left\{ v \in L^2(\Omega) : D^\alpha v \in L^2(\Omega), \forall 0 \leq |\alpha| \leq s \right\},$$

with the associated Sobolev norm

$$||v||_{H^s} = \left(\sum_{0 \leq |\alpha| \leq s} ||D^\alpha v||^2_{L^2(\Omega)} \right)^{\frac{1}{2}},$$

and the associated Sobolev seminorm

$$|v|_{H^s} = ||\nabla^s v||_{L^2(\Omega)} = \left(\sum_{|\alpha| = s} ||D^\alpha v||^2_{L^2(\Omega)} \right)^{\frac{1}{2}}.$$

The Sobolev spaces are defined by

$$H^s_0(\Omega) = \left\{ v \in H^s(\Omega) : v|_{\partial\Omega} = 0 \right\}.$$

In order to deal with a partition ξ_h of Ω, we define the broken Sobolev spaces by

$$H^s(\xi_h) = \left\{ v \in L^2(\Omega) : v|_K \in H^s(K), \forall K \in \xi_h \right\},$$

with the associated broken Sobolev norm

$$||v||_{H^s(\xi_h)} = \left(\sum_{K \in \xi_h} ||v||^2_{H^s(K)} \right)^{\frac{1}{2}},$$

and the associated broken seminorm

$$|v|_{H^0(\xi_h)} = \left(\sum_{K \in \xi_h} ||\nabla v||^2_{L^2(K)} \right)^{\frac{1}{2}}.$$

2.1.2 Trace Theorems

Theorem 2.1 (Theorem 2.5 [30]). *For $s_0 > 1/2$ and $s_1 > 3/2$, there exist trace operators $\gamma_0 : H^{s_0}(\Omega) \to H^{s_0 - 1/2}(\partial\Omega)$ and $\gamma_0 : H^{s_1}(\Omega) \to H^{s_1 - 1/2}(\partial\Omega)$ being extensions of the boundary values and boundary normal derivatives, respectively, with polygonal boundary $\partial\Omega$, and for $v \in C^1(\bar{\Omega})$, we have*

$$\gamma_0 v = v|_{\partial\Omega}, \qquad \gamma_1 v = \nabla v \cdot \vec{n}|_{\partial\Omega}.$$

2.1.2.1 Green's Theorem

Theorem 2.2. *Let Ω be a domain in $\mathbb{R}^2$, with boundary $\partial\Omega$ and exterior unit normal $\vec{n}$. Then, for all $v \in H^2(\Omega)$ and $w \in H^1(\Omega)$,*

$$-\int_\Omega \Delta u w dx = \int_\Omega \nabla u \nabla w dx - \int_{\partial\Omega} \vec{n} \cdot \nabla u w ds.$$

We consider the semi-linear ADR equations of the form

$$\begin{aligned} u_t - \epsilon\Delta u + \vec{b}\nabla u + r(u) &= f && \text{in } \Omega \subset \mathbb{R}^2, \\ u(x,t) &= g^D && \text{on } \Gamma_D, \\ \epsilon\nabla u(x,t) \cdot \vec{n} &= g^N && \text{on } \Gamma_N, \\ u(x,0) &= u_0 && \text{in } \Omega, \end{aligned} \tag{2.1}$$

with $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. In the above equation, $f \in L^2(\Omega)$ is the function of source contribution, ϵ is the constant of the diffusivity, $\vec{b} \in (W^{1,\infty}(\Omega))^2$ is the velocity field, $g^D \in H^{3/2}(\Gamma_D)$ is the Dirichlet boundary condition, $g^N \in H^{1/2}(\Gamma_N)$ is the Neumann boundary condition, $u_0 \in L^2(\Omega)$ is the initial condition and $\vec{n}$ denote outward normal vector to the boundary.

Let the mesh $\xi_h = \{K\}$ be a family of shape regular elements for some positive constant h_0

$$\max_{K\in\xi_h} \frac{h_K^2}{|K|} \leq h_0, \tag{2.2}$$

where $|K|$ and h_K denote the area and the diameter of the element K, respectively. Let also that $\bar{\Omega} = \cup\bar{K}$ and $K_i \cap K_j = \emptyset$ for $K_i, K_j \in \xi_h$. The set of interior domain, Neumann boundary and the Dirichlet boundary edges are denoted by Γ_h^0, Γ_h^N and Γ_h^D, respectively. $\Gamma_h^0 \cup \Gamma_h^D \cup \Gamma_h^N$ forms the outline of the mesh. For any $K \in \xi_h$, let $\mathbb{P}_k(K)$ be the set of all polynomials of degree at most k on K.

In order to discretize convection part of the problem (2.1) we will apply upwinding [26, 29]. Thus, let us decompose the boundary edges into the set Γ^+ of outflow edges and the set Γ^- of inflow edges defined by

$$\Gamma_h^- = \left\{ x \in \partial\Omega : \vec{b} \cdot \vec{n} < 0 \right\}, \qquad \Gamma_h^+ = \partial\Omega \setminus \Gamma_h^-,$$

where $\vec{n}$ is the unit normal vectors that point outward of the boundary $\partial\Omega$. Similarly, the set of outflow and inflow boundary edges of an element $K \in \xi_h$ is defined by

$$\partial K^- = \left\{ x \in \partial K : \vec{b} \cdot \vec{n}_K < 0 \right\}, \qquad \partial K^+ = \partial K \setminus \partial K^-,$$

where $\vec{n}_K$ is the unit normal vectors that point outward of the element boundary ∂K. Additionally, on an interior edge ∂K, we denote the trace of a function v from outside the element K by v^{out} and from inside the element K by v^{in}.

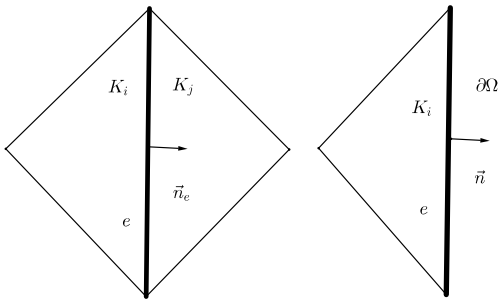

Figure 2.1: Two elements sharing an edge (left); an element near to domain boundary (right).

We set the finite dimensional solution and test function space by

$$V_h = \left\{ v \in L^2(\Omega) : v|_K \in \mathbb{P}_k(K), \forall K \in \xi_h \right\} \not\subset H_0^1(\Omega).$$

There are two different traces coming from the adjacent elements due to V_h contain discontinuous functions along the inter-element boundaries. Let $K_i, K_j \in \xi_h$ $(i < j)$ be two elements near an interior edge $e = K_i \cap K_j \subset \Gamma_h^0$ (cf Fig:2.1). We denote the trace of a scalar function v from insideK_i by v_i and from inside K_j by v_j. Then, we set the jump and average values of v on the edge e

$$[v] = v_i \vec{n}_e - v_j \vec{n}_e, \qquad \{v\} = \frac{1}{2}(v_i + v_j), \tag{2.3}$$

where $\vec{n}_e$ is the unit normal to the edge e oriented from K_i to K_j. Similarly, after setting the jump and average values of a vector valued function $\vec{q}$ has the form

$$[\vec{q}] = \vec{q}_i \cdot \vec{n}_e - \vec{q}_j \cdot \vec{n}_e, \qquad \{\vec{q}\} = \frac{1}{2}(\vec{q}_i + \vec{q}_j). \tag{2.4}$$

We also set,

$$\begin{aligned} [\vec{q}] &= \vec{q}_i \cdot \vec{n}, & \{\vec{q}\} &= \vec{q}_i, \\ [v] &= v_i \vec{n}, & \{v\} &= v_i, \end{aligned}$$

along any boundary edge $e = K_i \cap \partial\Omega$, where $\vec{n}$ is the unit outward normal to the boundary at e.

2.1.3 Construction of IPG Methods

Now, for constructing SIPG method, we multiply the continuous Equation (2.1) by a test function $v \in V_h$, we integrate over Ω ,

$$\sum_{K\in\xi_h}\int_K \frac{\partial u_h}{\partial t}v_h dx - \sum_{K\in\xi_h}\int_K \epsilon\Delta u_h v_h dx + \sum_{K\in\xi_h}\int_K \vec{b}\nabla u_h v_h dx + \sum_{K\in\xi_h}\int_K r(u_h)v_h dx = \sum_{K\in\xi_h}\int_K f v_h dx.$$

Applying the divergence theorem on every element integral gives

$$\sum_{K\in\xi_h}\int_K \frac{\partial u_h}{\partial t}v_h dx + \sum_{K\in\xi_h}\int_K \epsilon\nabla u_h\nabla v_h dx - \sum_{K\in\xi_h}\int_{\partial K} \epsilon(\nabla u_h\cdot\vec{n})v_h ds + \sum_{K\in\xi_h}\int_K \vec{b}\nabla u_h v_h dx - \sum_{K\in\xi_h}\int_K \vec{b}\cdot\vec{n}(u_h^{out}-u_h^{in})v_h ds - \sum_{K\in\xi_h}\int_K \vec{b}\cdot\vec{n}u_h^{in}v_h ds + \sum_{K\in\xi_h}\int_K r(u_h)v_h dx = \sum_{K\in\xi_h}\int_K f v_h dx.$$

It is easy to verify that $[\epsilon v\nabla u] = \{\epsilon\nabla u\}\cdot[v] + [\epsilon\nabla u]\{v\}$. Then, using also the fact that $[\nabla u] = 0$ (u is assumed to be smooth enough so that ∇u is continuous) and adding following equalities via $[u] = 0$ on the interior edges in order to handle the coercivity of the left hand side and control the jump terms,

$$\sum_{e\in\Gamma_h^0\cup\Gamma_h^D}\int_e \{\epsilon\nabla v_h\}\cdot[u_h]ds = \sum_{e\in\Gamma_h^D}\int_e g^D(\nabla v_h\cdot n)ds,$$

$$\sum_{e\in\Gamma_0\cup\Gamma_D}\frac{\sigma}{h_e}\int_e [u_h]\cdot[v_h]ds = \sum_{e\in\Gamma_D}\frac{\sigma}{h_e}\int_e g^D v_h ds,$$

we get,

$$\sum_{K\in\xi_h}\int_K \epsilon\nabla u_h\cdot\nabla v_h dx + \sum_{e\in\Gamma_0\cup\Gamma_D}\frac{\sigma\epsilon}{h_e}\int_e [u_h]\cdot[v_h]ds - \sum_{e\in\Gamma_0\cup\Gamma_D}\int_e \{\epsilon\nabla u_h\}\cdot[v_h]ds + \kappa\sum_{e\in\Gamma_0\cup\Gamma_D}\int_e \{\epsilon\nabla v_h\}\cdot[u_h]ds + \sum_{K\in\xi_h}\int_K \vec{\beta}\cdot\nabla u_h v_h dx + \sum_{K\in\xi_h}\int_{\partial K^-\setminus\partial\Omega}\vec{\beta}\cdot\vec{n}(u_h^{out}-u_h^{in})v_h ds - \sum_{K\in\xi_h}\int_{\partial K^-\cap\Gamma^-}\vec{\beta}\cdot\vec{n}u_h^{in}v_h ds + \sum_{K\in\xi_h}\int_K \alpha u_h v_h dx = \sum_{K\in\xi_h}\int_K f v_h dx + \sum_{e\in\Gamma_D}\int_e g^D\left(\frac{\sigma\epsilon}{h_e}v_h + \kappa\epsilon\nabla v_h\cdot\mathbf{n}\right)ds - \sum_{K\in\xi_h}\int_{\partial K^-\cap\Gamma^-}\vec{\beta}\cdot\vec{n}g^D v_h ds + \sum_{e\in\Gamma_N}\int_e g^N v_h ds,$$

so that we obtained the IPG formulation. The sort of the interior penalty Galerkin method depends on κ in. Depending on the coefficient $\kappa = -1, 0, 1$ the method varies as, Symmetric interior penalty Galerkin (SIPG) method, Incomplete interior penalty Galerkin (IIPG) method, Non-symmetric interior penalty Galerkin (NIPG) method, respectively.

Finally, we can give SIPG with upwinding formulation of Equation (2.1) as: find $u_h \in V_h$ such that

$$\int_\Omega \frac{\partial u_h}{\partial t} v_h dx + a_h(u_h, v_h) = l_h(v_h), \qquad \forall v_h \in V_h, \tag{2.5}$$

with coreesponding bilinear form

$$\begin{aligned}
a_h(u_h, v_h) = &\sum_{K\in\xi_h} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx + \sum_{e\in\Gamma_0\cup\Gamma_D} \frac{\sigma\epsilon}{h_e} \int_e [u_h]\cdot[v_h] ds \\
&- \sum_{e\in\Gamma_0\cup\Gamma_D} \int_e \{\epsilon\nabla u_h\}\cdot[v_h] ds + \kappa \sum_{e\in\Gamma_0\cup\Gamma_D} \int_e \{\epsilon\nabla v_h\}\cdot[u_h] ds \\
&\sum_{K\in\xi_h} \int_K \vec{\beta}\cdot\nabla u_h v_h dx + \sum_{K\in\xi_h} \int_{\partial K^-\setminus\partial\Omega} \vec{\beta}\cdot\vec{n}(u_h^{out} - u_h^{in}) v_h ds \\
&- \sum_{K\in\xi_h} \int_{\partial K^-\cap\Gamma^-} \vec{\beta}\cdot\vec{n} u_h^{in} v_h ds + \sum_{K\in\xi_h} \int_K \alpha u_h v_h dx, \\
l_h(v_h) = &\sum_{K\in\xi_h} \int_K f v_h dx + \sum_{e\in\Gamma_D} \int_e g^D \left(\frac{\sigma\epsilon}{h_e} v_h + \kappa\epsilon\nabla v_h\cdot\mathbf{n}\right) ds \\
&- \sum_{K\in\xi_h} \int_{\partial K^-\cap\Gamma^-} \vec{\beta}\cdot\vec{n} g^D v_h ds + \sum_{e\in\Gamma_N} \int_e g^N v_h ds.
\end{aligned}$$

2.1.4 Forming The Linear Systems

The discrete DG scheme for elliptic problems with bilinear form is given as [36]

$$a_h(u_h, v_h) := D_h(u_h, v_h) + C_h(u_h, v_h) + R_h(u_h, v_h) = l_h(v_h), \tag{2.6}$$

where the forms $D_h(u_h, v_h), C_h(u_h, v_h)$ and $R_h(u_h, v_h)$ are corresponding to the diffusion,advection and linear reaction parts of the problem, respectively, given by [36]

$$D_h(u_h, v_h) = \sum_{K\in\xi_h} \int_K \epsilon \nabla u_h \cdot \nabla v_h dx + \sum_{e\in\Gamma_0\cup\Gamma_D} \frac{\sigma\epsilon}{h_e} \int_e [u_h]\cdot[v_h] ds \tag{2.7}$$
$$- \sum_{e\in\Gamma_0\cup\Gamma_D} \int_e \{\epsilon\nabla u_h\}\cdot[v_h] ds + \kappa \sum_{e\in\Gamma_0\cup\Gamma_D} \int_e \{\epsilon\nabla v_h\}\cdot[u_h] ds,$$

$$C_h(u_h, v_h) = \sum_{K\in\xi_h} \int_K \vec{\beta}\cdot\nabla u_h v_h dx \tag{2.8}$$
$$+ \sum_{K\in\xi_h} \int_{\partial K^-\setminus\partial\Omega} \vec{\beta}\cdot\vec{n}(u_h^{out} - u_h^{in}) v_h ds$$
$$- \sum_{K\in\xi_h} \int_{\partial K^-\cap\Gamma^-} \vec{\beta}\cdot\vec{n} u_h^{in} v_h ds,$$

$$R_h(u_h, v_h) = \sum_{K\in\xi_h} \int_K \alpha u_h v_h dx, \tag{2.9}$$

$$l_h(v_h) = \sum_{K\in\xi_h} \int_K f v_h dx + \sum_{e\in\Gamma_D} \int_e g^D \left(\frac{\sigma\epsilon}{h_e} v_h + \kappa\epsilon\nabla v_h\cdot\mathbf{n}\right) ds \tag{2.10}$$
$$- \sum_{K\in\xi_h} \int_{\partial K^-\cap\Gamma^-} \vec{\beta}\cdot\vec{n} g^D v_h ds + \sum_{e\in\Gamma_N} \int_e g^N v_h ds.$$

The discrete solution $u_h \in V_h$ has the form

$$u_h = \sum_{j=1}^{N} v_j \phi_j, \tag{2.11}$$

with a set of basis functions $\{\phi_i\}_{i=1}^N$ spanning the space V_h and $v = (v_1, v_2, \ldots, v_N)^T$ is the unknown coefficient vector. After substituting (2.11) into (2.6) and taking $v_h = \phi_i$, we get for $i = 1, \ldots, N$, the linear systems of equations

$$\sum_{j=1}^{N} v_j D_h(\phi_j, \phi_i) + \sum_{j=1}^{N} v_j C_h(\phi_j, \phi_i) + \sum_{j=1}^{N} v_j R_h(\phi_j, \phi_i) = l_h(\phi_i). \tag{2.12}$$

To form the linear system in matrix-vector form, for $i = 1, \ldots, N$, we need the matrices $D, C, R \in \mathbb{R}^{N\times N}$ related to the terms including the forms D_h, C_h and R_h in Equation (2.12), respectively, satisfying

$$Dv + Cv + Rv = F, \tag{2.13}$$

with the unknown coefficient vector v and the vector $F \in \mathbb{R}^N$ related to the linear functionals $l_h(\phi_i)$ such that $F_i = l_h(\phi_i)$, $i = 1, \ldots, N$.

To solve non-linear problems, we need the vector$H \in \mathbb{R}^N$ related to the non-linear term such that

$$H_i(v) = \int_\Omega r\left(\sum_{j=1}^{N} v_j \phi_j\right) \phi_i dx, \qquad i = 1, \ldots, N. \tag{2.14}$$

After substituting (2.14) into (2.12) get for $i = 1, \ldots, N$ the non-linear systems of equations

$$\sum_{j=1}^{N} v_j D_h(\phi_j, \phi_i) + \sum_{j=1}^{N} v_j C_h(\phi_j, \phi_i) + \sum_{j=1}^{N} v_j R_h(\phi_j, \phi_i) + \int_\Omega r\left(u_h\right) \phi_i dx = l_h(\phi_i), \tag{2.15}$$

which can be written in the matrix-vector form of

$$Dv + Cv + Rv + H(v) = F, \tag{2.16}$$

where, the matrices $D, C, R \in \mathbb{R}^{N \times N}$ and the vector $H, F \in \mathbb{R}^N$.

For parabolic problems we will rewrite Equation 2.16 as

$$Mv + Sv + H(v) = F, \tag{2.17}$$

where, the matrix $S = (D + C + R)$ is the stiffness matrix and $M \in \mathbb{R}^{N \times N}$ is the symmetric positive definite mass matrix which by DG construction has a symmetric block diagonal structure [36].

CHAPTER 3

TIME DISCRETIZATION and SPLITTING

In this chapter we give an overview for the Rosenbrock methods and time splitting techniques for the efficient solution of the semi-linear ADR equations.

3.1 Rosenbrock Methods

For stiff ODEs arising by discretizing PDEs, explicit methods have the disadvantage that they require small time step sizes for a stable numerical solutions which increase the elapsed time of computation. On the other hand implicit schemes produces numerically stable solutions, but at each time step a large linear system have to be dealt, which is in many cases ill-conditioned. We consider here the Rosenbrock methods which avoids the solution of nonlinear systems, working with the exact Jacobian [11, 18]. Rosenbrock methods are derived as a special cases of diagonally implicit Runge-Kutta methods in 1963 [19]. It is known that Rosenbrock methods affected from order reduction dealing with stiff ODEs. Various type of Rosenbrock methods with multiple stages were developed [23, 24] which not suffer from order reduction. These methods are used for solving large systems of nonlinear ODEs, differential algebraic equations (DAEs) and nonlinear parabolic PDEs efficiently.

We consider the following initial value problem arising from the dG discretization of the semi-linear ADR equation

$$Mu' = F(u), \qquad u(0) = u_0, \tag{3.1}$$

where M is an invertible (N, N)-matrix and u_0 is initial condition. While dealing with Galerkin methods we will assume that M is the symmetric positive definite mass matrix.

The Rosenbrock methods were derived from the simple idea that subtract a linear autonomous term Ju from both sides of Equation (3.1),

$$Mu' - Ju = F(u) - Ju, \tag{3.2}$$

and discretize the left-hand part implicitly, but the right-hand part explicitly. This leads structural advantage compared to implicit methods. Depending on Ju two variants exist:

- Methods with exact Jacobian matrix '$J = F'(u)$'. This type of variant is used in Rosenbrock-Wanner (ROW) methods. The main disadvantage of the ROW-method is that Jacobian matrix must be computed at every integration step which makes it less attractive for integrating large systems.
- Methods with inexact Jacobian matrix '$J \approx F'(u)$'. This type of variant is used in W-methods. The choice of Jacobian matrix is relatively free for W-methods. Thus, W-methods lead computational advantages since the Jacobian matrix is not evaluated at every step.

3.1.1 Structure of Rosenbrock Methods

Rosenbrock methods are class of linearly implicit Runge-Kutta methods and derived by linearizing the diagonally implicit Runge-Kutta (DIRK) scheme.

At first to consider differential equations in autonomous form

$$u' = F(u), \qquad u(0) = u_0, \tag{3.3}$$

a nonlinear DIRK scheme is given by [18]

$$k_i = hF\left(u_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j + \alpha_{ii} k_i\right), \qquad i = 1, \ldots, s, \tag{3.4}$$

where the solution at the next time step is given by

$$u_{n+1} = u_n + \sum_{i=1}^{s} b_i k_i. \tag{3.5}$$

Equation (3.4) is linearised around

$$g_i = u_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j, \tag{3.6}$$

become

$$k_i = hF(g_i) + hF'(g_i)\alpha_{ii} k_i. \tag{3.7}$$

For acceleration of the computations, the Jacobian $F'(g_i)$ is replaced by $J_F = F'(u_0)$ so that the Jacobian will not be needed at every stage of Rosenbrock computation.

Thus an s-stage Rosenbrock method [18] reads as:

$$k_i = hF\left(u_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + hJ_F \sum_{j=1}^{i-1} \gamma_{ij} k_j, \qquad i = 1, \ldots, s, \tag{3.8}$$

$$u_{n+1} = u_n + \sum_{i=1}^{s} b_i k_i.$$

with the coefficients α_{ij}, γ_{ij} and b_i which are generally shown in a Butcher tableau.

The nonlinear Equations (3.8) require the solution of a linear system with the matrix $I - h\gamma_{ii}$ and the matrix-vector multiplication $J_F \sum \gamma_{ij} k_j$. In order to avoid this multiplication we introduce the new variables:

$$U_i = \sum_{j=1}^{i-1} \gamma_{ij} k_j, \qquad i = 1, \ldots, s. \tag{3.9}$$

If $\gamma_{ij} \neq 0$ for $i \leq j$, then the matrix $\Gamma = (\gamma_{ij})$ is invertible and k_i can be determined from U_i with

$$k_i = \frac{1}{\gamma_{ii}} U_i - \sum_{j=1}^{i-1} c_{ij} U_j, \tag{3.10}$$

where C is given by

$$C = diag(\gamma_{11}^{-1}, \ldots, \gamma_{ss}^{-1}) - \Gamma^{-1}. \tag{3.11}$$

So Rosenbrock method can be implemented as :

$$\left(\frac{I}{h\gamma_{ii}} - J_F\right) U_i = F\left(u_n + \sum_{j=1}^{i-1} a_{ij} U_j\right) + \sum_{j=1}^{i-1} \frac{c_{ij}}{h} U_j, \tag{3.12}$$

where $a_{ij} = \alpha_{ij}\Gamma^{-1}$, $(m_1, \ldots, m_s) = (b_1, \ldots, b_s)\Gamma^{-1}$ and u_{n+1} is given by

$$u_{n+1} = u_n + \sum_{i=1}^{s} m_j U_j. \tag{3.13}$$

Non-autonomous problems

$$y' = F(u, t), \tag{3.14}$$

can be converted to autonomous form by adding $t' = 1$. Then the augmented system(3.8) become

$$\begin{aligned} k_i = hF\left(t_0 + \alpha_i h, u_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + \gamma_i h^2 \frac{\partial F}{\partial x}(t_0, u_0) \\ + h \frac{\partial F}{\partial u}(t_0, u_0) \sum_{j=1}^{i} \gamma_{ij} k_j, \\ u_{n+1} = u_n + \sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{3.15}$$

where the additional coefficients are given by

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \qquad \gamma_i = \sum_{j=1}^{i-1} \gamma_{ij}. \tag{3.16}$$

Implicit differential equations in the form of

$$My' = F(u,t), \tag{3.17}$$

with a constant matrix M can be converted to autonomous form by multiplying Equation (3.17) with M^{-1}, applying method in Equation (3.15), and then multiplying the resulting formula with M we obtain

$$\begin{aligned} Mk_i = hF\left(t_0 + \alpha_i h, u_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + \gamma_i h^2 \frac{\partial F}{\partial x}(t_0, u_0) \\ + h\frac{\partial F}{\partial u}(t_0, u_0) \sum_{j=1}^{i} \gamma_{ij} k_j, \\ u_{n+1} = u_n + \sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{3.18}$$

In this chapter, we consider Rosenbrock methods of different order with two, three and four stages designed for efficient solution of nonlinear ODEs.

The second order 2-stage ROS2 method for autonomous ODE systems is applied to atmospheric dispersion problems in [38]. For non-autonomous systems 3.17 the scheme can be written as

$$\begin{aligned} \left(M - \gamma\tau \frac{\partial F}{\partial u}(t_n, u_n)\right) k_1 &= F(t_n, u_n) + \gamma\tau \frac{\partial F}{\partial t}(t_n, u_n), \\ \left(M - \gamma\tau \frac{\partial F}{\partial u}(t_n, u_n)\right) k_2 &= F(t_{n+1}, u_n + \tau k_1) - 2Mk_1 \\ &\quad - \gamma\tau \frac{\partial F}{\partial t}(t_n, u_n), \\ u_{n+1} &= u_n + \frac{3}{2}\tau k_1 + \frac{1}{2}\tau k_2. \end{aligned} \tag{3.19}$$

After substituting Equation (3.19) in Equation (2.17) fully discrete formulation of Equation 2.1 with 2-stage ROS2 method in time and SIPG in space takes the form

$$\begin{aligned} \left(M - \gamma\tau \left(S + J_H(u_n)\right) k_1 &= -Su_n - H(u_n) + \tilde{F}(\cdot, t_n) + \tau\gamma\partial_t F(t_n, u_n) \right. \\ \left(M - \gamma\tau \left(S + J_H(u_n)\right) k_2 &= -S(u_n + \tau k_1) - H(u_n + \tau k_1) + \tilde{F}(\cdot, t_{n+1}) \\ &\quad - 2Mk_1 - \tau\gamma\partial_t F(t_n, u_n), \\ u_{n+1} &= u_n + \frac{3}{2}\tau k_1 + \frac{1}{2}\tau k_2, \end{aligned}$$

where the jacobian of $F(t,u)$ is

$$J_F = -S - J_H(u), \tag{3.20}$$

and $J_H(u)$ is the jacobian of non-linear part in $H(u)$.

Table 3.1: Coefficients for the 3-stage ROS3P method.

a_{21} = 1.267949192431123e+ 00	c_{21} = -1.607695154586736e+00
a_{31} = 1.267949192431123e+00	c_{31} = -3.464101615137755e+00
a_{32} = 0.000000000000000e+00	c_{32}= -1.732050807568877e+00
α_1 = 0.000000000000000e+00	γ_1= 7.886751345948129e-01
α_2 = 1.000000000000000e+00	γ_2 = -2.113248654051871e-01
α_3 = 1.000000000000000e+00	γ_3= -1.077350269189626e+00
m_1 = 2.000000000000000e+00	γ = 7.886751345948129e-01
m_2 = 5.773502691896258e-01	
m_3= 4.226497308103742e-01	

The ROS2 method have proven to be very effective in many applications, e.g. atmospheric dispersion problems [38], chemical systems [39], atmospheric multiphase chemical kinetics [12], geothermal processes [35].

We consider the third order 3-stage ROS3P [24] method. For avoiding matrix-vector multiplication we will use following notation. For initial value problem

$$\partial_t u = F(u,t), \quad u(0) = u_0, \quad 0 < t \leq T, \tag{3.21}$$

3-stage ROS3P method with the step size $\tau > 0$ has the form [24]

$$\left(\frac{I}{\tau\gamma} - \partial_u F(t_n, u_n)\right) U_{ni} = F(t_n + \alpha_i\tau, u_n + \sum_{j=1}^{i-1} a_{ij}U_{nj}) + \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} U_{nj} + \tau\gamma_i \partial_t F(t_n, u_n), \tag{3.22}$$

$$u_{n+1} = u_n + \sum_{i=1}^{3} m_i U_{ni}\ , \tag{3.23}$$

for $i = 1, 2, 3$.

The coefficients of ROS3P method are presented in Table 3.1.

After substituting Equation (3.22) in Equation (2.17) fully discrete formulation of 2.1 with 3-stage ROS3P method in time SIPG in space takes the form

$$\left(\frac{M}{\tau\gamma} + S + J_H(u_n)\right) U_{ni} = - S(u_n + \sum_{j=1}^{i-1} a_{ij}U_{nj}) - H(u_n + \sum_{j=1}^{i-1} a_{ij}U_{nj}) + \tilde{F}(\cdot, t_n + \alpha_i\tau)$$
$$+ \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} MU_{nj} + \tau\gamma_i\partial_t F(\cdot, t_n),$$
$$u_{n+1} = u_n + \sum_{i=1}^{3} m_i U_{ni} \ ,$$

where $i = 1, 2, 3$, the jacobian of $F(t, u)$ is

$$J_F = -S - J_H(u), \tag{3.24}$$

and $J_H(u)$ is the jacobian of non-linear part in $H(u)$.

The 4-stage third order L-stable ROS3PL-method for non-autonomous systems with the recursive form [23] can be written as

$$\left(\frac{M}{\tau\gamma} - \partial_u F(t_n, u_n)\right) U_{ni} = F(t_n + \alpha_i\tau, u_n + \sum_{j=1}^{i-1} a_{ij}U_{nj}) \tag{3.25}$$
$$- \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} MU_{nj} + \tau\gamma_i\partial_t F(t_n, u_n),$$
$$u_{n+1} = u_n + \sum_{i=1}^{3} m_i U_{ni},$$

where $i = 1, \ldots, 4$ and coefficients are presented in Table 3.2. After substituting Equation (3.25), in Equation (2.17) fully discrete formulation of Equation 2.1 with 4-stage ROS3PL method in time SIPG in space takes the form

$$\left(\frac{M}{\tau\gamma} + S + J_H(u_n)\right) U_{ni} = -S(u_n + \sum_{j=1}^{i-1} a_{ij}U_{nj}) - H(u_n + \sum_{j=1}^{i-1} a_{ij}U_{nj}) + \tilde{F}(\cdot, t_n + \alpha_i\tau)$$
$$- \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} MU_{nj} + \tau\gamma_i\partial_t F(\cdot, t_n),$$
$$u_{n+1} = u_n + \sum_{i=1}^{3} m_i U_{ni},$$

where $i = 1, \ldots, 4$, the jacobian of $F(t, u)$ is

$$J_F = -S - J_H(u), \tag{3.26}$$

and $J_H(u)$ is the jacobian of non-linear part in $H(u)$.

Table 3.2: Coefficients for the 4-stage ROS3PL method.

γ = 0.4358665215084590	
a_{11} = 0.000000000000000	c_{11} = 2.294280360279042
a_{21} = 1.147140180139521	c_{21} = 2.631861185781065
a_{22} = 0.000000000000000	c_{22} = 2.294280360279042
a_{31} = 2.463070773030053	c_{31} = 1.302364158113095
a_{32} = 1.147140180139521	c_{32} = -2.769432022251304
a_{33} = 0.000000000000000	c_{33}= 2.294280360279042
a_{41} = 2.463070773030053	c_{41}= 1.552568958732400
a_{42} = 1.147140180139521	c_{42} = -2.587743501215153
a_{43} = 0.000000000000000	c_{43}= 1.416993298352020
a_{44} = 0.000000000000000	c_{44}= 2.294280360279042
γ_1 = 0.435866521508459	α_1 = 0.000000000000000
γ_2 = -0.064133478491541	α_2 = 0.500000000000000
γ_3 = 0.111028172512505	α_3 = 1.000000000000000
γ_4 = 0.000000000000000	α_4 = 1.000000000000000
m_1 = 2.463070773030053	
m_2 = 1.147140180139521	
m_3= 0.000000000000000	
m_4 = 1.000000000000000	

3.2 Operator Splitting Methods

Operator splitting methods are well studied field in the numerical solution of ordinary differential equations. Generally the main idea is to split the differential operator into several parts, where each part represents a particular physical phenomenon, such as convection, diffusion. Operator splitting is an attractive technique, since complex equation system maybe split into simpler parts that are easier to solve. Operating splitting methods not only used for simplifying the complexity. Different constructions of splitting methods give arise to the higher order methods. After splitting one can treat each part of the original operator independently and can lead to very efficient methods.

Different versions of operator splitting as with various time integrators are applied for solving semi-linear parabolic ADR equations, see for example [7, 22, 25, 38].

For the illustration of the operator splitting methods, we consider the form of

$$\frac{\partial u(t)}{\partial t} = Au(t) + Bu(t), \qquad t \in [0, T],\ u(0) = u_0, \tag{3.27}$$

where u_0 is given and A and B linear bounded operators in the Banach-space. Solution of Equation (3.27) at the time t is $u(t) = e^{(A+B)t}u_0$. An alternative solution can be done by replacing the Equation (3.27) with the subproblems on the subintervals:

$$\frac{\partial u^*(t)}{\partial t} = Au^*(t), \qquad t \in \left(t^n, t^{n+1}\right),\ u^*(t^n) = u^n_{sp}, \tag{3.28}$$

$$\frac{\partial u^{**}(t)}{\partial t} = Bu^{**}(t), \qquad t \in \left(t^n, t^{n+1}\right),\ u^{**}(t^n) = u^*(t^{n+1}), \tag{3.29}$$

whereby $u^0_{sp} = u_0$ is initial condition. This operator splitting technique is called sequential operator-splitting. It is the simplest operator-splitting method. Clearly, splitting the original Equation (3.27) in the form of the subproblems in Equations (3.28)-(3.29) causes error, called local splitting error. For sequential operator-splitting method local splitting error can be derived as follows [14]:

$$\rho_n = \frac{1}{\tau_n}\left(\exp(\tau_n(A+B)) - \exp(\tau_n B)\exp(\tau_n A)\right)u^n_{sp} \tag{3.30}$$

$$= \frac{1}{2}\tau_n[A, B]u(t^n) + \mathcal{O}(\tau_n^2), \tag{3.31}$$

where $\tau_n = t^{n+1} - t^n$. We define $[A, B] := AB - BA$ as the commutator of A and B. Thus, when the operators commute, then the method is exact.

An idea for constructing accurate schemes which may be used with large step size can be construction of higher order methods from its numerical map Φ_t. So Φ_t of Equation (3.27) satisfies

$$\Phi_t = e^{(A+B)t} + \mathcal{O}(t^{p+1}) \tag{3.32}$$

with the order p. For higher order operator splitting methods a standard technique is

composing Φ_t from more than two exponentials. Such as

$$e^{(A+B)t} = \prod_{i=1}^{m} e^{a_i t A} e^{b_i t B} + \mathcal{O}(t^{m+1}), \tag{3.33}$$

where A, B are noncommutative operators, a_i, b_i are real numbers and t is time step.

Several compositions can be cast on the free parameters $a_1, \ldots, a_m$, and $b_1, \ldots, b_m$, for determining conditions.

For example Strang splitting method for Equation (3.27) can be cast into the general form Equation (3.33) with

$$m = 2,\ a_1 = a_2 = \frac{1}{2}, \qquad\qquad b_1 = 1,\ b_2 = 0. \tag{3.34}$$

So numerical solution take the form

$$u(t) = e^{A\frac{t}{2}} e^{Bt} e^{A\frac{t}{2}}. \tag{3.35}$$

In this study, we consider a very famous operator splitting method called Strang-Marchuk splitting. This idea of splitting was proposed by Strang [34] and Marchuk (1971). It is generally called as Strang splitting. Strang splitting is generally used to accelerate computations for problems containing operators on very different time scales, for instance, chemical reactions in fluid dynamics, and to solve multidimensional partial differential equations. Strang splitting is second order method so that the method is both provide accuracy and efficiency.

3.2.1 Strang Splitting

Strang splitting method is pointed it out as a popular and commonly used operator splitting method. Strang splitting algorithm is as follows :

$$\frac{\partial u^*(t)}{\partial t} = Au^*(t), \quad t^n \le t \le t^{n+1/2}, \qquad u^*(t^n) = u^n_{sp}, \tag{3.36}$$

$$\frac{\partial u^{**}(t)}{\partial t} = Bu^{**}(t), \quad t^n \le t \le t^{n+1}, \qquad u^{**}(t^n) = u^*(t^{n+1/2}), \tag{3.37}$$

$$\frac{\partial u^{***}(t)}{\partial t} = Au^{***}(t), \ \ t^{n+1/2} \le t \le t^{n+1}, \qquad u^{***}(t^{n+1/2}) = u^{**}(t^n), \tag{3.38}$$

where $t^{n+1/2} = t^n + \dfrac{\tau}{2}$ and $u^n_{sp} = u_0$, and the approximation for the next time step t^{n+1} is defined as $u^n_{sp} = u^{***}(t^{n+1})$. Moreover, the idea behind Strang splitting can be summarized in Figure 3.1.

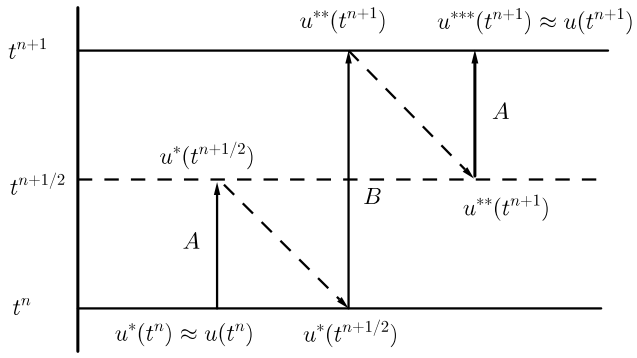

Figure 3.1: Systematic schema of Strang splitting method.

3.2.2 ROS2 within Strang-type operator splitting

In this part of our study we will illustrate Strang-type operator splitting method with ROS2 and explicit trapezoid rule.

Assume that the following equation denotes the ODE system obtained from spatial discretization

$$u_t = F(u,t) \equiv F_T(u,t) + F_R(u,t), \qquad F_T(u,t) \equiv F_A(u,t) + F_D(u,t), \tag{3.39}$$

where the vector function F_T is supposed to contain the semidiscrete transport contributions from advection and diffusion, represented here by F_A and F_D, respectively. Likewise, F_R is supposed to contain the reaction and source part.

Application of Strang splitting to Equation (3.39) takes the form

$$u_t = F_T(u,t), \qquad t^n \le t \le t^{n+1/2}, \tag{3.40}$$
$$v_t = F_R(v,t), \qquad t^n \le t \le t^{n+1}, \tag{3.41}$$
$$w_t = F_T(w,t), \qquad t^{n+1/2} \le t \le t^{n+1}. \tag{3.42}$$

We will apply explicit trapezoid rule to Equations (3.40), (3.42) and ROS2 to Equation

(3.41). Thus we will consider the combination [38]

$$
\begin{aligned}
&U_0 = u_n, \qquad (3.43)\\
&U_1 = U_0 + \frac{1}{4}\tau F_T(U_0, t_n) + \frac{1}{4}\tau F_T\left(U_0 + \frac{1}{2}\tau F_T(U_0, t_n), t_{n+1/2}\right),\\
&U_2 = U_1 + \frac{3}{2}\tau k_1 + \frac{1}{2}\tau k_2,\\
&U_3 = U_2 + \frac{1}{4}\tau F_T(U_2, t_n) + \frac{1}{4}\tau F_T\left(U_2 + \frac{1}{2}\tau F_T(U_2, t_n), t_{n+1/2}\right),\\
&u_{n+1} = U_3,
\end{aligned}
$$

where k_1 and k_2 for non autonomous are given by [6]

$$\left(I - \gamma\tau\frac{\partial F_R}{\partial u}(u^n, t^n)\right) k_1 = F_R(u^n, t^n) + \gamma\tau\frac{\partial F_R}{\partial t}(u^n, t^n), \qquad (3.44)$$

$$\left(I - \gamma\tau\frac{\partial F_R}{\partial u}(u^n, t^n)\right) k_2 = F_R(u^n + \tau k_1, t^{n+1}) - 2k_1 - \gamma\tau\frac{\partial F_R}{\partial t}(u^n, t^n). \qquad (3.45)$$

Substituting Equation (2.16) in Equation (3.43) with $F_R(u,t) = -Ru - H(u) + \tilde{F}(t)$ and $F_T(u,t) = Du - Cu$ we have

$$
\begin{aligned}
Mu^{n+1/2} &= Mu^n + \frac{1}{4}\tau F_T(u^n, t^n)\\
&\quad + \frac{1}{4}\tau F_T\left(u^n + \frac{1}{2}\tau F_T(u^n, t^n), t^{n+1/2}\right), \qquad (3.46)\\
v^{n+1} &= v^n + \frac{3}{2}\tau k_1 + \frac{1}{2}\tau k_2, \qquad (3.47)\\
Mw^{n+1} &= Mw^{n+1/2} + \frac{1}{4}\tau F_T(w^{n+1/2}, t^{n+1/2})\\
&\quad + \frac{1}{4}\tau F_T\left(w^{n+1/2} + \frac{1}{2}\tau F_T(w^{n+1/2}, t^{n+1/2}), t^{n+1}\right), \qquad (3.48)
\end{aligned}
$$

where u_0 is initial condition, $u_{n+1/2} = v_n$,$v_{n+1} = w_{n+1/2}$ and k_1, k_2 are given by

$$\left(M - \gamma\tau\frac{\partial F_R}{\partial u}(u^n, t^n)\right) k_1 = F_R(u^n, t^n) + \gamma\tau\frac{\partial F_R}{\partial t}(u^n, t^n), \qquad (3.49)$$

$$\left(M - \gamma\tau\frac{\partial F_R}{\partial u}(u^n, t^n)\right) k_2 = F_R(u^n + \tau k_1, t^{n+1}) - 2Mk_1 - \gamma\tau\frac{\partial F_R}{\partial t}(u^n, t^n). \qquad (3.50)$$

CHAPTER 4

NUMERICAL RESULTS

In this chapter, we have shown the performance of the Rosenbrock time integrators with different time splitting under dG spatial discretization for some semi-linear ADR equations. We have shown the theoretically predicted order of convergence in space for SIPG discretization of ROS2, ROS3P and ROS3PL time integrators. For spatial discretization we used DGFEM package proposed in [37]. All numerical examples are taken from [9]. In order to find more efficent solution of the semi-linear ADR equations we applied Strang splitting to the problems with second order Rosenbrock method ROS2 time integrator to the stiff parts of the equations. We have shown that this combination is more efficent then backward Euler within Strang splitting as mentioned in [38]. In the last problem of this chapter, we compared ROS2 within Strang splitting with ROS3P and we have shown that the latter is less efficent then the first one. We have compared the norm preconditioner with ILU preconditioner for the iterative solution of linear systems arising from SIPG discretization with GMRES method.

4.1 Numerical Results

4.1.1 Test Example-1

Our first problem is a nonlinear convection–diffusion-reaction problem with exact solution

$$\begin{aligned}
&\frac{\partial u}{\partial t} - \Delta u + v_1\frac{\partial u}{\partial x} + v_2\frac{\partial u}{\partial y} + ku + \frac{u^3}{1+u^2} = f(x,y,t) \quad \forall (x,y,t) \in \Omega \times [0,5],\\
&u(x,0,t) = u(x,1,t) = 0 \quad \forall x \in [0,1] \text{ and } \forall t \in [0,5],\\
&u(0,y,t) = u(1,y,t) = 0 \quad \forall y \in [0,1] \text{ and } \forall t \in [0,5],\\
&u(x,y,0) = x(1-x)y(1-y) \quad \forall (x,y) \in \Omega,
\end{aligned}$$

with $\Omega = [0,1] \times [0,1]$, $v_1 = 2 - x$, $v_2 = (1+y)(1+e^{-t})$, $k(x,y,t) = 1 + xye^{-t}$ and the source term $f(x,y,t)$ is chosen adequately to obtain as exact solution $u(x,y,t) = e^{-t}x(1-x)y(1-y)$.

In Table 4.3 , 4.4 and Table 4.1 we show the spatial errors of SIPG discretization for ROS3P and ROS3PL with $\Delta t = 0.001$. The errors and observed order of convergence

Table 4.1: Spatial errors and corresponding order of convergence of ROS3P method for $\Delta t = 0.001$ and linearly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates	$L^2(H^1)$ Error	$L^2(H^1)$ Rates
96	1/4	2.0337e-03	-	3.4001e-02	-
384	1/8	5.7143e-04	1.832	1.7098e-02	0.992
1536	1/16	1.4837e-04	1.945	8.5153e-03	1.006
6144	1/32	3.6243e-05	2.033	4.2428e-03	1.005

of Example-1 using SIPG for spatial discretization and ROS3P for time discretization are displayed in Table 4.1. The observed spatial orders of convergence measured in H^1 -norm reveal is 1 and in L^2-norm it is 2. ROS3P is third order and A-stable method and for quadratic elements as in Table 4.2 one can observe that orders of convergence measured in L^2 -norm reveal is 3 and in the global H^1 -norm it is 2. The numerical order of convergence for SIPG with quadratic elements confirm the theoretical orders of convergence.

Table 4.2: Spatial errors and corresponding order of convergence of ROS3P method for $\Delta t = 0.0001$ and quadraticly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates	$L^2(H^1)$ Error	$L^2(H^1)$ Rates
192	1/4	1.257e-04	-	5.070e-03	-
768	1/8	1.526e-05	3.0415	1.288e-03	1.9768
3072	1/16	1.893e-06	3.0114	3.229e-04	1.9960
12288	1/32	4.008e-07	2.2399	8.081e-05	1.9986

In Table 4.3 we displayed the spatial errors and order of convergence for Example-1 with SIPG and ROS3PL using linear elements. The observed spatial order of convergence for the global L^2-norm is 2 and for the global H^1-norm is 1. ROS3PL is a L-stable method so one can see that using large time steps does not reduce the order of convergence as in Table 4.4. For quadratic elements with time step $\Delta t = 0.0001$ and $\Delta x = 1/32$ we observed order reduction in ROS3P while with time step $\Delta t = 0.001$ ROS3PL is not affected from order reduction as it can be seen in Table 4.4.

Table 4.3: Spatial errors and corresponding order of convergence of ROS3PL method for $\Delta t = 0.001$ and linearly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates	$L^2(H^1)$ Error	$L^2(H^1)$ Rates
96	1/4	2.036e-03	-	3.400e-02	-
384	1/8	5.737e-04	1.827	1.710e-02	0.992
1536	1/16	1.507e-04	1.929	8.515e-03	1.006
6144	1/32	3.851e-05	1.968	4.243e-03	1.005

Temporal errors of ROS3P and ROS3PL for $\Delta x = 1/256$ are displayed in Table 4.5. Observed order of convergence for ROS3P is to 0.4 and for ROS3PL it is 1.

In Table 4.6 we observed that order of convergence of ROS2 method with $\Delta t = 0.01$ for the global L^2-norm is 2 and for the global H^1-norm is 1. ROS2 method is a L-stable

Table 4.4: Spatial errors and corresponding order of convergence of ROS3PL method for $\Delta t = 0.001$ and quadraticly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates	$L^2(H^1)$ Error	$L^2(H^1)$ Rates
96	1/4	1.25778e-04	-	5.07025e-03	-
384	1/8	1.53162e-05	3.0377	1.28801e-03	1.977
1536	1/16	1.89193e-06	3.0171	3.22788e-04	1.996
6144	1/32	2.37555e-07	2.9935	8.06698e-05	2.000

Table 4.5: Temporal errors and corresponding order of convergence of ROS3P and ROS3PL for $\Delta x = 1/64$ and linearly discretized SIPG method.

		ROS3P		ROS3PL	
DoFs	Δt	$L^2(L^2)$ Error	Rate	$L^2(L^2)$ Error	Rate
24576	1.000	8.861e-04	-	8.610e-04	-
24576	0.500	2.216e-04	2.000	4.393e-04	0.971
24576	0.250	1.164e-04	0.928	2.285e-04	0.943
24576	0.125	9.148e-05	0.348	1.096e-04	1.060

2-step model which is fast and allows us for using large time step Δt. ROS3PL and ROS2 are L-stable methods but we note that ROS2 which is second order method converge to its classical order with linearly discretized SIPG by the global L^2-norm while ROS3PL which is third order method converge to its classical order with quadraticly discretized SIPG by the global L^2-norm.

Table 4.6: Spatial errors and corresponding order of convergence of ROS2 for $\Delta t = 0.01$ and linearly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates	$L^2(H^1)$ Error	$L^2(H^1)$ Rates
96	1/4	2.038e-03	-	3.39906e-02	-
384	1/8	5.754e-04	1.824	1.70880e-02	0.992
1536	1/16	1.523e-04	1.918	8.50944e-03	1.006
6144	1/32	4.010e-05	1.925	4.23991e-03	1.005

In Table 4.7 we displayed the temporal errors of ROS2 and ROS2 within Strang splitting for linearly disceretized SIPG method. The observed values for ROS2 is 1.6 while for ROS2 within Strang it is 1.

Table 4.7: Strang splitting and ROS2 temporal errors for $\Delta x = 1/32$ and linearly disceretized SIPG method. For ROS2 within Strang splitting advection and diffusion part solved with ROS2 rest of the equation solved with explicit trapezoid rule.

		ROS2		ROS2 within Strang	
DoFs	Δt	$L^2(L^2)$ Error	Rate	$L^2(L^2)$ Error	Rate
6144	1.000	7.793e-03	-	1.855e-01	-
6144	0.500	3.121e-03	1.320	1.083e-01	0.776
6144	0.250	1.056e-03	1.564	5.698e-02	0.927
6144	0.125	3.329e-04	1.665	2.824e-02	1.013

In Table 4.8 we have presented spatial error and convergence order of Example-1 where the advection and diffusion part solved by ROS2 and rest of the equation solved by explicit trapezoid rule. In Table 4.9 we presented spatial error and convergence order of Example-1 by backward Euler within Strang splitting.

An application to atmospheric dispersion problems has been done in [38], the authors pointed out that stability of Strang splitting depends on very small time steps. The authors found out that if solution of the stiff part is solved by ROS2 and rest of the equation solved by explicit trapezoid rule, the splitting allows to use larger time steps. They applied ROS2 within Strang splitting to a advection-diffusion reaction equation. Similarly we observed that for $\Delta t = 0.001$ and $\Delta x = 1/32$ order of convergence of ROS2 within Strang splitting is 1.157 while the observed order of convergence of backward Euler is -0.402. This shows that backward Euler within Strang splitting starts to diverge at $\Delta x = 1/32$ while ROS2 within Strang splitting still converging. Moreover, we presented time splitting errors for different time steps of backward Euler within Strang splitting and ROS2 within Strang splitting in the global L^2-norm in Figure 4.1 , Figure 4.2 and Figure 4.3 . We observed that as the time step decrease splitting errors of ROS2 within Strang is decreasing while backward Euler is increasing. For instance the global L^2 error of backward Euler within Strang in Figure 4.3 for $\Delta x = 1/16$ and $\Delta x = 1/32$ is respectively 2.986e-03 and 3.083e-03 while for ROS2 case it is 1.505e-03 and 1.498e-03. This shows that for small time steps backward Euler is not stable while ROS2 is still stable. Thus we found out that this time splitting strategy is efficent for Strang splitting.

Table 4.8: Strang splitting spatial errors and corresponding order of convergence for linearly discretized SIPG method with $\Delta t = 0.001$. Advection and diffusion solved with ROS2 rest of the equation solved with explicit trapezoid rule.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rate
96	1/4	2.021e-03	-
384	1/8	5.612e-04	1.849
1536	1/16	1.481e-04	1.922
6144	1/32	6.639e-05	1.157

Table 4.9: Strang splitting spatial errors and corresponding order of convergence for linearly discretized SIPG method with $\Delta t = 0.001$. All parts of the splitting schemes solved by Backward Euler.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rate
96	1/4	1.927e-03	-
384	1/8	4.647e-04	2.052
1536	1/16	8.070e-05	2.526
6144	1/32	1.067e-04	-0.402

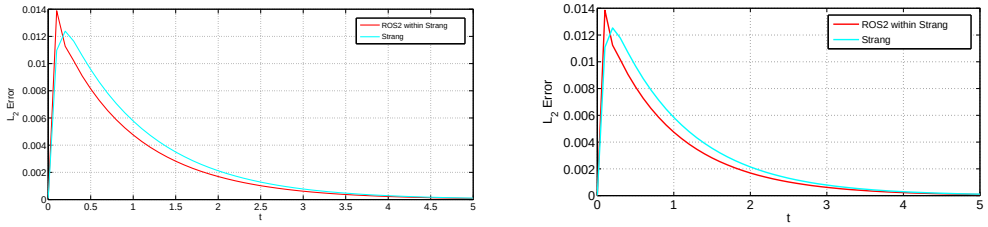

Figure 4.1: Global L_2 error of backward Euler and ROS2 within Strang splitting at time t for $\Delta t = 0.1$. $\Delta x = 1/16$ (left) , $\Delta x = 1/32$ (right).

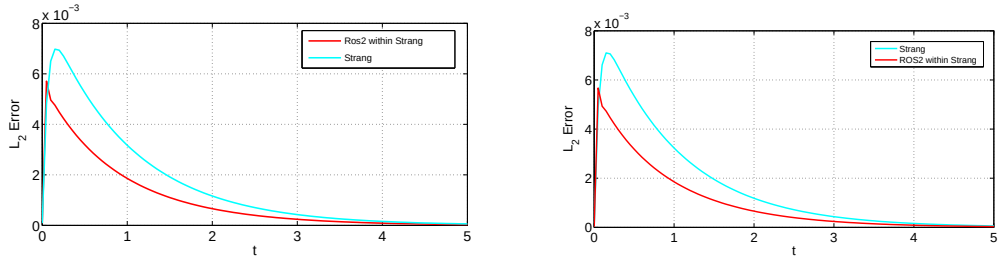

Figure 4.2: Global L_2 error of backward Euler and ROS2 within Strang splitting at time t for $\Delta t = 0.05$. $\Delta x = 1/16$ (left) , $\Delta x = 1/32$ (right).

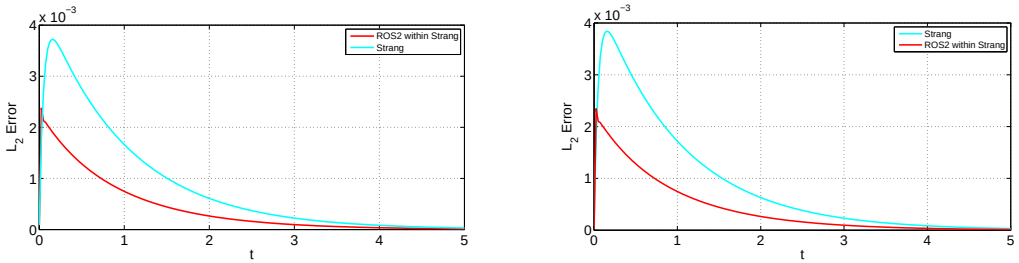

Figure 4.3: Global L_2 error of backward Euler and ROS2 within Strang splitting at time t for $\Delta t = 0.025$. $\Delta x = 1/16$ (left) , $\Delta x = 1/32$ (right).

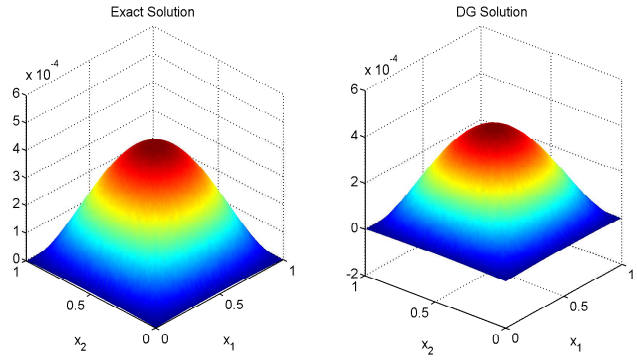

Figure 4.4: Solution of the Example-1 using ROS2 within Strang splitting scheme for time integration and (SIPG) for spatial discretization.

In Figure 4.4 we displayed the exact solution and numerical solution of Example-1 using (SIPG) for spatial discretization and ROS2 within Strang splitting as a time integration. In the Figure $\Delta x = 1/16$, $\Delta t = 0.01$ and 1536 degree of freedoms used.

4.1.2 Test Example-2

Our second problem is a nonlinear ADR problem without exact solution.It has the form of

$$\begin{aligned}
&\frac{\partial u}{\partial t} - \Delta u + v_1\frac{\partial u}{\partial x} + v_2\frac{\partial u}{\partial y} + ku + ue^{-u^2} = f(x,y,t) \quad \forall (x,y,t) \in \Omega \times [0,5],\\
&u(x,0,t) = u(x,1,t) = 0 \quad \forall x \in [0,1] \text{ and } \forall t \in [0,5],\\
&u(0,y,t) = u(1,y,t) = 0 \quad \forall y \in [0,1] \text{ and } \forall t \in [0,5],\\
&u(x,y,0) = x^2(1-x)^2y^2(1-y)^2 \quad \forall (x,y) \in \Omega,
\end{aligned}$$

where $\Omega = [0,1] \times [0,1]$, $v_1 = 1 + xy, v_2 = 1 + x$, $k(x,y,t) = 1 + (x+y)^2e^{-t}$and the source term $f(x,y,t) = 10^4t^2e^{-t}h(x)h(y)$ with $h(\xi) = e^{-\xi} + e^{\xi-1} - (1+e^{-1})$.

We have estimated the numerical errors by using the double mesh principle by

$$E_{N,\tau} = \max_{\substack{(x_i,y_j)\in\Omega_{1/N} \\ t_m = m\tau, m=1,2,3,\ldots,\frac{5}{\tau}}} |U^{N,\tau}(x_i,y_j,t_m) - U^{2N,\tau}(x_i,y_j,t_m)|,$$

where $U^{2N,\tau}(x_i,y_j,t_m)$ is the numerical solution obtained by $2N \times 2N$.

Table 4.10: ROS3P spatial errors and corresponding order of convergence for $\Delta t = 0.01$ and linearly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates
96	1/4	-	-
384	1/8	8.034e+00	-
1536	1/16	2.091e+00	1.942
6144	1/32	5.283e-01	1.985
24576	1/64	1.324e-01	1.996

In Table 4.10 we displayed order of convergence and spatial errors of ROS3P method with $\Delta t = 0.01$ measured in the global L^2-norm. We observed that comparing with previous problem larger time steps not changed the stability.

Table 4.11: ROS2 within Strang splitting spatial errors and corresponding order of convergence for $\Delta t = 0.01$ and linearly discretized SIPG method.

DoFs	Δx	$L^2(L^2)$ Error	$L^2(L^2)$ Rates
96	1/4	-	-
384	1/8	9.510e+00	-
1536	1/16	2.484e+00	1.937
6144	1/32	6.282e-01	1.983
24576	1/64	1.575e-01	1.996

In Table 4.10 and Table 4.11 we observed that even though ROS2 is second order and

ROS3P is third order method for linearly discretized SIPG method ROS2 converge to its classical order while ROS3P has order reduction for Example-2.

4.1.3 Test Example-3

In this case we have considered an initial boundary value problem of the form:

$$\begin{aligned}
&\frac{\partial u}{\partial t} - \epsilon\Delta u + ku + u^3 = f(x,y,t) \quad \forall (x,y,t) \in \Omega \times [0,5],\\
&u(x,0,t) = u(x,1,t) = 0 \quad \forall x \in [0,1] \text{ and } \forall t \in [0,5],\\
&u(0,y,t) = u(1,y,t) = 0 \quad \forall y \in [0,1] \text{ and } \forall t \in [0,5],\\
&u(x,y,0) = x^2(1-x)^2y^2(1-y)^2 \quad \forall (x,y) \in \Omega,
\end{aligned}$$

where $\Omega = [0,1] \times [0,1]$,$\epsilon > 0$,$k(x,y,t) = 1 + (x+y)^2 + e^{-t}$and the source term $f(x,y,t) = e^{-t}x(1-x)y(1-y)$.Also this problem has no exact solution.

Table 4.12: Spatial errors and order of convergence of ROS3P for $\Delta t = 0.01$ and $\epsilon = 1$.

DoFs	Δx	$L^2(L^2)$ Error	Rates
96	1/4	-	-
384	1/8	4.396e-03	-
1536	1/16	1.112e-03	1.983
6144	1/32	2.799e-04	1.990
24576	1/64	7.010e-05	1.997

Spatial errors and order of convergence of Example-3 are presented in Table 4.12. Observed order of convergence measured in the global L^2-norm is 2. Different from previous 2 example this example has no convection term. Numerical errors measured same as in Example-2.

Table 4.13: Spatial errors and order of convergence of ROS2 within Strang splitting for $\Delta t = 0.01$ and $\epsilon = 1$.

DoFs	Δx	$L^2(L^2)$ Error	Rates
96	1/4	-	-
384	1/8	5.684e-03	-
1536	1/16	1.464e-03	1.957
6144	1/32	3.688e-04	1.989
24576	1/64	9.239e-05	1.997

As in previous example, In Table 4.12 and Table 4.13 we observed that even though ROS2 is second order and ROS3P is third order method for linearly discretized SIPG method ROS2 converge to its classical order while ROS3P has order reduction for Example-3.

Table 4.14: Spatial errors and order of convergence of ROS2 within Strang splitting and ROS3P for $\Delta t = 0.1$ and $\epsilon = 1e-9$.

		ROS3P		ROS2 within Strang	
DoFs	Δx	$L^2(L^2)$ Error	Rates	$L^2(L^2)$ Error	Rates
96	1/4	-	-	-	-
384	1/8	1.867e-02	-	1.880e-02	-
1536	1/16	4.791e-03	1.962	4.816e-03	1.965
6144	1/32	1.220e-03	1.973	1.211e-03	1.991
24576	1/64	3.222e-04	1.921	3.033e-04	1.998

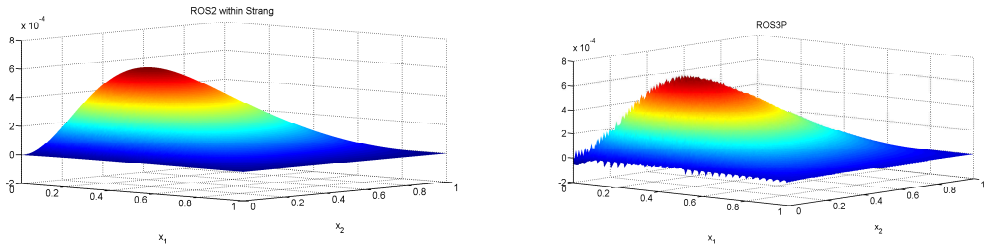

Figure 4.5: Solution of the Example-3 using ROS2 within Strang splitting and ROS3P for time integration and (SIPG) for spatial discretization with $\Delta t = 0.1$ $\Delta x = 1/64$ and $\epsilon = 1e-9$. ROS2 within Strang splitting (left), ROS3P (right).

In Table 4.14, we observed that for $\epsilon = 1e-9$ and $\Delta x = 1/64$ for the case of ROS3P order reduction has appeared while for the case of ROS2 within Strang splitting still it is converging to its classical order. Moreover, order reduction also can be observed from Figure 4.5.

4.2 Preconditioning

Numerical solution of the large linear systems of the form $Ax = b$ with large condition number $\kappa(A)$ is an issue since it takes too much time or it has lack of accuracy. There are two numerical solvers for such system: direct and iterative. Direct solvers usually faster then iterative solvers for small linear systems. Also there is no need to think about the preconditioners. However, these solvers needs too much memory and CPU time for larger systems. On the other hand, iterative solvers do not need too much memory and they are able to solve very large linear systems. Also their other advantage is that they often parallize well. Only disvantage of iterative solvers is choice of the solver and preconditioner. Because efficiency of iterative solver depends on good preconditioner. In this thesis we are dealing with linear systems arising from spatial discretization of dG methods which leads to large and dense linear systems where the condition number $\kappa(A)$ of the matrix A increases as $h \to 0$. Hence, to obtain an efficient solution of the linear systems one must use efficient methods such as the generalized minimum residual method (GMRES). An alternative to GMRES for solving the linear system $Ax = b$ is the conjugate gradient method however if A is not symmetric-positive-definite (SPD), it cannot be directly applied. On the other hand, for such linear systems using only GMRES does not guarantee an efficent solution so that we used a preconditioner which have been found efficent for linear systems arising from dG spatial discretization [16]. We have compared the norm preconditoner with incomplete LU factorization (ILU) preconditioner and we have shown that relative residual of the norm preconditioner decreases in remarkably less iteration then ILU preconditioner.

The generalized minimum residual method (GMRES) is

$$x^{n+1} = x^0 + \sum_{i=0}^{n} \alpha_i A^i r^0,$$

where $r^0 = b - Ax^0$ and $\alpha_i \in \mathbb{R}$ are chosen so that

$$\begin{aligned} R(x^{n+1}) &= \min_y R(y), \\ y &\in x^0 + K_{n+1}, \\ \mathcal{K}_{n+1} &= \{z | z = \sum_{i=0}^{n} c_i A^i r^0, \;\; c_i \in \mathbb{R}\}. \end{aligned}$$

The idea behind the GMRES method is based on solving a least squares problem at each step of the iteration. The approximate solution is given by a vector $x^n \in \mathcal{K}_n$ (the n-th order Krylov subspace) such that the residual

$$||r^n||_2 = ||Ax^n - b||_2 \tag{4.1}$$

is minimized. In order to solve this least squares problem, one can use the Arnoldi iteration to construct a sequence of Krylov matrices then solve it iteratively.

4.2.1 Preconditioners

Spectral properties of the coefficient matrix determines the convergence rate of iterative methods. Hence one attempt to get same solution with better spectral properties may be transforming the linear system into one that is equivalent. From this point of view, a preconditioner is a matrix that performs such a transformation.

Suppose we wish to solve a $m \times m$ linear system

$$Ax = b. \tag{4.2}$$

By multiplying both side of equation with inverse of any nonsingular $m \times m$ matrix M , the system

$$M^{-1}Ax = M^{-1}b \tag{4.3}$$

has the same solution. Suppose we want to solve 4.3 iteratively then the convergence rate will depend on the matrix $M^{-1}A$ instead of matrix A. In order to solve 4.2 more rapidly the matrix M must be well chosen.

There is no general theory on efficient selection of preconditioners. A good preconditioner M expected to be close enough to matrix A. By close enough to matrix A we mean the eigenvalues of $M^{-1}A$ are close to 1 and $||M^{-1}A - I||_2$ is small. Another property of a good preconditioner is easiness of solution. Preconditioned system in Equation 4.3 must be easier to solve than in Equation 4.2.

There are several ideas on preconditioning, one of them is *diagonal scaling or Jacobi*. The idea behind diagonal scaling is to find a diagonal matrix M that minimizing the condition number of the matrix $M^{-1}A$. One example to this kind of preconditioning is $M = diag(A)$. For some problems, this transformation is satisfactory for rapidness of the convergence.

Another popular idea called as *Incomplete Cholesky or LU factorization.* This preconditioning idea was famous in the 1970s. Let us consider a sparse A matrix with just a few nonzeros per row. The disadvantage of the methods such as Gaussian elimination or Cholesky factorization is that these processes decrease the number of zeros,so that if $A = RR^T$, then the matrix R will not be very sparse. Instead, we seek to find decomposition as $A \approx \tilde{R}\tilde{R}^T$ where $\tilde{R}$ allowed to have nonzeros only in positions where A has nonzeros so that we can conserve the sparsity.

The last idea that we mention is called *Norm-Preconditioning.* This idea is based on norm-equivalence. Moreover, the idea behind the method is to find a precondtioner which is norm-equivalent to symmetric positive definite matrix. An application of norm-preconditioning for linear systems arising from dG has been done in [16]. The author claimed that $A_s = (A + A^T)/2$ is a good preconditioner for the linear systems of dG.

We have given the idea of preconditioning with left preconditioners in previous section. Another idea for preconditioning of the linear system of Equation 4.2 is to transforming

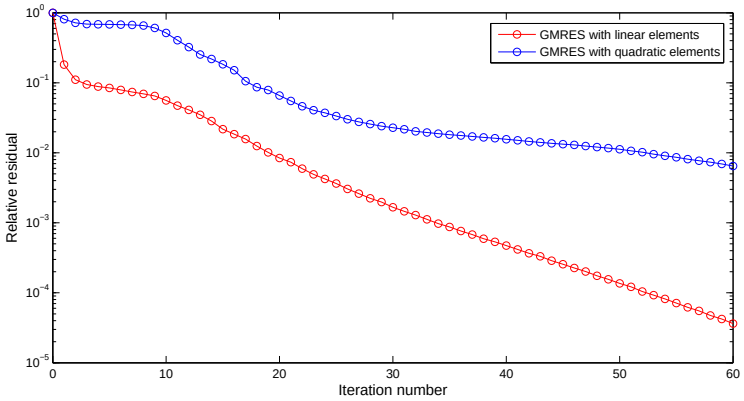

Figure 4.6: Semilogarithmic plot of relative residual of GMRES method on linearly and quadraticly discretized SIPG Method without preconditioner

the system to

$$AM^{-1}u = b, \tag{4.4}$$

where $u = Mx$.The initial residual is $r^0 = b - Ax^0 = b - AM^{-1}u^0$ so note that u^0 is not necessary to start the algorithm.

We have combined Strang splitting with Rosenbrock methods in previous chapter. Relative residual of linearly and quadraticly discretized SIPG method of the first linear system arising from ROS2 within Strang splitting without preconditioner is demonstrated in Figure 4.6. Computation time of the linearly discretized system took 0.844216 seconds with Intel-Core i3 processor and 4GB RAM. At 60th iteration relative residual have become 8.8648e-04. For quadraticly discretized system observed elapsed time for the solution of the unpreconditioned GMRES is 0.682956 seconds and at final iteration observed relative residual is 6.46559e-03. We note that linearly discretized system has 24576 degree of freedoms while quadraticly discretized system has 12288 degree of freedoms, altough, solution of linearly discretized system is less expensive then quadraticly discretized system.

In Figure 4.7 we have demonstrated relative residual of ILU and norm preconditioned linear system both for linearly and quadraticly discretized case. The linear system here is same with the system of Figure 4.6. We have used $A_s = (A + A^T)/2$ which suggested in [16] as a norm preconditioner. ILU and norm preconditioned system are remarkably faster converged to the tolerance then the unprecontioned system. In 60 iteration ILU have not reached the desired solution while norm preconditioned system got only 6 iteration for linear and quadratic case. We note that tolerance of ILU in Figure 4.7 is 1e-8.

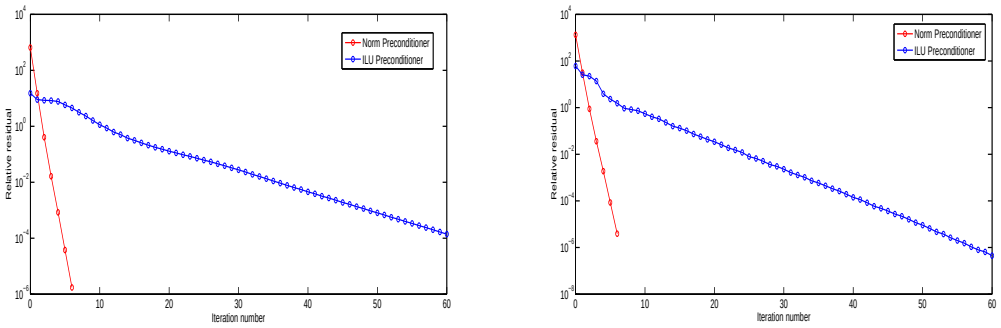

Figure 4.7: Semilogarithmic plot of relative residual of GMRES method on SIPG Method with ILU and Norm preconditioner. Linearly discretized (left), Quadraticly discretized (right).

Computation time of the linearly discretized system with ILU and Norm preconditioner is respectively 0.991449 and 0.739987 seconds. Observed relative errors for ILU and Norm preconditioner is respectively 9.1606e-06 and 2.6560e-09. For linearly and quadraticly discretized SIPG case, norm preconditioners have been found significantly efficient.

Table 4.15: Condition numbers of stiffness matrix for linearly and quadraticly discretized SIPG with different penalty terms σ and $\Delta x = 1/16$.

	σ	10	20	30	40
Linear Elements	Unpreconditioned	3.8146e+03	7.9976e+03	1.2314e+04	1.6630e+04
	ILU preconditioner	1.2655e+03	2.5243e+03	3.8807e+03	5.0948e+03
	Norm preconditioner	4.8117	4.7848	4.7851	4.7852
Quadratic Elements	Unpreconditioned	9.3235e+03	2.0087e+04	3.2201e+04	4.4534e+04
	ILU preconditioner	1.6686e+03	4.5976e+03	7.1879e+03	9.6445e+03
	Norm preconditioner	5.2023	5.1860	5.1828	5.1815

In order to obtain a coercive bilinear form of SIPG the penalty parameter σ must be chosen large enough [30]. In Table 4.15, we have displayed condition numbers of stiffness matrix obtained from SIPG method for different choice of the penalty parameters. The condition number increase for unpreconditioned linear systems and remain constant for the ILU and norm preconditioners. Norm preconditioner reduces the condition number dramatically.

CHAPTER 5

CONCLUSIONS

In this thesis, we have studied semi-linear advection-diffusion-reaction equations. We have discretized the space with symmetric interior penalty Galerkin (SIPG) method to deal with the unphysical oscillations due to the advection term. We have demonstrated spatial and temporal errors and corresponding order of convergence for Rosenbrock methods and ROS2 within Strang splitting for three different problems. Computational results obtained by SIPG discretization in space confirms the theoretical orders of Rosenbrock methods.

In conclusion, we have obtained theoretically experimental convergence rates of Rosenbrock methods for ADR equations numerically. We conclude that ROS2 within Strang splitting more efficent candidate then ROS3P for stiff problems. Moreover, we concluded that ROS2 is good candidate for dealing with the stiff equations. Also we conclude that the norm preconditioner for linear systems arising from SIPG method is more efficent then ILU preconditioner.

REFERENCES

[1] P. F. Antonietti and E. Süli, Domain decomposition preconditioning for discontinuous Galerkin approximations of convection-diffusion problems, in *Domain decomposition methods in science and engineering XVIII*, pp. 259–266, Springer, 2009.

[2] D. Arnold, F. Brezzi, B. Cockborn, and L. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal., 39, pp. 1749–1779, 2002.

[3] B. Ayuso and L. D. Marini, Discontinuous Galerkin methods for advection-diffusion-reaction problems, SIAM J. Numer. Anal., 47, pp. 1391–1420, 2009.

[4] I. Babuška and M. Zlámal, Nonconforming elements in the finite element method with penalty, SIAM J. Numer. Anal., 10, pp. 45–59, 1973.

[5] C. E. Baumann and J. T. Oden, A discontinuous hp finite element method for convection-diffusion problems, Comput. Methods Appl. Mech. Engrg., 175, pp. 311–341, 1999.

[6] P. Benner and H. Mena, Rosenbrock methods for solving riccati differential equations, Automatic Control, IEEE Transactions on, 58(11), pp. 2950–2956, Nov 2013, ISSN 0018-9286.

[7] J. Blom and J. Verwer, A comparison of integration methods for atmospheric transport-chemistry problems, Journal of Computational and Applied Mathematics, 126(1–2), pp. 381 – 396, 2000.

[8] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo, Discontinuous Galerkin approximations for elliptic problems, Numer. Methods Partial Differential Equations, 16, pp. 365–378, 2000.

[9] B. Bujanda and J. Jorge, Efficient linearly implicit methods for nonlinear multidimensional parabolic problems, Journal of Computational and Applied Mathematics, 164–165, pp. 159 – 174, 2004.

[10] B. Cockburn and C. W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal., 35, pp. 2440–2463, 1999.

[11] P. Deuflhard and M. Weiser, *Adaptive Numerical Solution of PDEs*, De Gruyter Textbook, Walter de Gruyter, 2012.

[12] R. Djouad, B. Sportisse, and N. Audiffren, Numerical simulation of aqueous-phase atmospheric models: use of a non-autonomous rosenbrock method, Atmospheric Environment, 36(5), pp. 873–879, 2002.

[13] J. Dougles and T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in R. Glowinski and J. L. Lions, editors, *Computing Methods in Applied Sciences*, volume 58 of *Lecture Notes in Phys*, pp. 207–216, Springer Berlin Heidelberg, 1976.

[14] J. Geiser, *Decomposition methods for differential equations: theory and applications*, CRC Press, 2009.

[15] E. H. Georgoulis, Discontinuous Galerkin methods for linear problems: An introduction, in E. H. Georgoulis, A. Iske, and J. Levesley, editors, *Approximation Algorithms for Complex Systems*, volume 3 of *Springer Proceedings in Mathematics*, pp. 91–126, Springer Berlin Heidelberg, 2011.

[16] E. H. Georgoulis and D. Loghin, Krylov-subspace preconditioners for discontinuous galerkin finite element methods, 2006.

[17] M. Günther and M. Hoschek, Row methods adapted to electric circuit simulation packages, Journal of computational and applied mathematics, 82(1), pp. 159–170, 1997.

[18] E. Hairer, S. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Lecture Notes in Economic and Mathematical Systems, Springer, 1993, ISBN 9783540604525.

[19] H.H.Rosenbrock, Some general implicit processes for the numerical solution of differential equations, The Computer Journal, 5–4, pp. 329–330, 1963.

[20] P. Houston, C. Schwab, and E. Süli, Discontinuous hp-finite element methods for advection-diffusion-reaction problems, SIAM J. Numer. Anal., 39, pp. 2133–2163, 2002.

[21] P. Kaps and G. Wanner, A study of Rosenbrock-type methods of high order, Numerische Mathematik, 38(2), pp. 279–298, 1981.

[22] K. H. Karlsen, K.-A. Lie, J. R. Natvig, H. F. Nordhaug, and H. K. Dahle, Operator splitting methods for systems of convection–diffusion equations: nonlinear error mechanisms and correction strategies, Journal of Computational Physics, 173(2), pp. 636–663, 2001.

[23] J. Lang and D. Teleaga, Towards a fully space-time adaptive FEM for magnetoquasistatics, Magnetics, IEEE Transactions on, 44(6), pp. 1238–1241, June 2008, ISSN 0018-9464.

[24] J. Lang and J. Verwer, ROS3P - an accurate third-order Rosenbrock solver designed for parabolic problems, Numer. Math, 41, pp. 731–738, 2001.

[25] D. Lanser and J. Verwer, Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling, Journal of Computational and Applied Mathematics, 111(1–2), pp. 201 – 216, 1999.

[26] P. Lesaint and P. A. Raviert, On a finite element for solving the neutron transport equation, mathematical aspects of finite elements in partial differential equations, Math. Res. Center, Univ. of Wisconsin-Madison, Academic Press, New York, pp. 89–123, 1974.

[27] J. Peraire and P. O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput., 30, pp. 1806–1824, 2008.

[28] P.-O. Piersson and J. Peraire, Newton -GMRES preconditioning for discontinuous galerkin discretizations of the navier-stokes equations, SIAM Journal on Scientific Computing, 30(6), pp. 2709–2733, 2008.

[29] W. H. Reed and T. R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alomos Scientific Laboratory, Los Alomos, NM, 1973.

[30] B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations, Theory and implementation*, SIAM, 2008.

[31] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing, 7(3), pp. 856–869, 1986.

[32] B. Schippmann and H. Burchard, Rosenbrock methods in biogeochemical modelling–a comparison to Runge–Kutta methods and modified Patankar schemes, Ocean Modelling, 37(3), pp. 112–121, 2011.

[33] L. Shampine and H. Watts, Block implicit one-step methods, Mathematics of Computation, 23(108), pp. 731–740, 1969.

[34] G. Strang, On the construction and comparison of difference schemes, SIAM Journal on Numerical Analysis, 5(3), pp. 506–517, 1968.

[35] A. Tambue, I. Berre, and J. M. Nordbotten, Efficient simulation of geothermal processes in heterogeneous porous media based on the exponential rosenbrock–euler and rosenbrock-type methods, Advances in Water Resources, 53, pp. 250–262, 2013.

[36] M. Uzunca, *Adaptive Discontinuous Galerkin Methods for Non-linear Reactive Flows*, Ph.D. thesis, Middle East Technical University, 2014.

[37] M. Uzunca and B. Karasözen, A matlab tutorial for diffusion-convection-reaction equations using DGFEM, arXiv preprint arXiv:1502.02941, 2015.

[38] J. G. Verwer, E. Spee, J. Blom, and W. Hundsdorfer, A second-order Rosenbrock method applied to photochemical dispersion problems, SIAM Journal on Scientific Computing, 20(4), pp. 1456–1480, 1999.

[39] D. A. Voss and A.-Q. M. Khaliq, Parallel Rosenbrock methods for chemical systems, Computers & chemistry, 25(1), pp. 101–107, 2001.