Doctoral Dissertation

Full Discretization Process for Advection—Diffusion—Reaction Equations

DOAN DUY HAI

JULY 2012

THE DEPARTMENT OF APPLIED PHYSICS
QUANTUM ENGINEERING DESIGN COURSE
GRADUATE SCHOOL OF ENGINEERING
OSAKA UNIVERSITY

Full Discretization Process for Advection–Diffusion–Reaction Equations

A DISSERTATION PRESENTED

BY

DOAN DUY HAI

TO

THE DEPARTMENT OF APPLIED PHYSICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE SUBJECT OF

QUANTUM ENGINEERING DESIGN COURSE

OSAKA UNIVERSITY

SUITA, OSAKA

JULY 2012

TITLE OF THESIS

Full Discretization Process for Advection–Diffusion–Reaction Equations

PREFACE

This thesis is intended to present a full discretization process to solving numerically advection-diffusion-reaction (ADR) equations and to show how to apply to various practical models. Our approach for discretization for ADR equations bases on method-of-lines (MOLs) which consists of two phases, spatial and temporal discretizations. The first phase is to discretize space variables and its result is a system of ordinary differential equations (ODEs). This system is numerically integrated in time variables afterward in the second phase of discretization process. Here we employ discontinuous Galerkin (DG) methods in the former and Rosenbrock strong stability-preserving ones in the latter phase.

The DG methods are able to be considered as an extension of classical finite element (FE) methods. The idea of FE methods in general and DG methods in particular is to divide computational domain into small pieces called elements and then to approximate the exact solution on each element by easy-to-compute functions (e.g. polynomials, wavelets). The main difference between theses two kind of methods is that the numerical solutions using the DG methods are allowed to be discontinuous element-to-element while they are required to be continuous on the whole domain if the classical FE methods is in used. Such discontinuities offer more degree of freedoms (DOFs) and therefore allow us more flexibility to design different discretization schemes for different terms in ADR equations. The main difficulty of using discontinuous functions in DG methods is how to transfer information such as fluxes in between elements. This task is quite trivial in classical FE methods because the numerical solution is continuous and hence the information is automatically shifted between elements. For DG methods, because of discontinuity, transferring information has to be done manually by carefully designing so-called numerical fluxes. With well-designed numerical fluxes, the DG methods are able to attain high order of accuracy and stability as well.

The spatial phase results a huge system of ODEs consisting of three discretized terms from advections, diffusion, and reaction. Each of these terms has completely different properties that require special treatment in the temporal phase. The discretized term corresponding to advection part as one will see is although non-stiff but containing in itself non-smooth operators. The non-stiffness and non-smoothness properties require

an explicit solver. Meanwhile, the discretized term associated with diffusion and reaction parts is smooth but very stiff. These facts mean that our solver must be explicit with the discretized advection term and be implicit with the rest while it should be stable enough to preserve the positivity of our problems and fast enough to be realistically applicable.

This thesis is divided into four chapters. The first chapter introduces ADR equations and several models that lead to such kinds of mathematical equations. It also mentions several numerical solvers dealing with such kind of problems so far and our motivation to propose a whole new discretization procedure for ADR equations. The second chapter is concerned with the spatial discretization. Several numerical fluxes for advection and diffusion equations are investigated in this chapter. The most suitable ones are chosen to put together and fulfill the first phase of our procedure. The third phase is completed in the third chapter. A new class of temporal integration methods with respect to special properties of the semi-discretized system obtained in the previous chapter is proposed. Numerical results of the new discretization procedure are given in the last chapter. By these results, the strength and shortcomings regarding efficiency, accuracy, and robustness of our methods are given.

Contents

Listing of figures

List of Tables

To Mom and Dad.

1 Introduction

The aim of this thesis is the development of full discretization procedure for advection-diffusion-reaction (ADR) equations. Difficulties in solving ADR equations are order of accuracy, stability, computer memory storage, computational speed, and robustness. This thesis is concerned with methods developed to prevail over these challenges.

1.1 Motivation

ADR equations are used to describe various important problems in physics, chemistry, biology, and engineering such as physicochemical hydrodynamical models [31], adsorbate induced phase transition model [23], and angiogenesis models [1]. Numerical solver for such kind of partial differential equations (PDEs) are usually based on an approach known as the method of lines (MOL) . For a given evolution PDE, the MOL involves first discretizing in spatial variables resulting in a system of ordinary differential equations (ODEs) called semi-discretized system. This system is then solved using a numerical ODE integration method.

For the first phase of MOL, there are several spatial discretization methods includ-

ing finite difference [35], finite volume [27], and finite element [36] methods. None of them is perfect. Finite difference methods (FDMs) are fast, high order of accuracy, and having deeply theoretical understanding. But they are inflexible with respect of boundary conditions and computational domain. For FDMs, imposing boundary conditions is not trivial task and sometimes too simple treatment with boundary conditions heavily damages numerical solutions. Finite volumes methods are robust but difficult to achieve high order on general grid due to extended stencils. Finite element methods are geometrically flexible and high order of accuracy but not well suited for problems with direction, e.g. advection-dominated ADR equations. Recently, a new class of spatial discretization methods called *discontinuous Galerkin (DG) methods* [22] is being developed. These methods base on classical finite element ones but they allow approximate solutions to be discontinuous between elements. Such discontinuities make them appropriate to problems with direction. Moreover these methods inherit all advantages of classical finite element methods such as high order of accuracy and geometrical flexibility. In this thesis, the DG methods are employed for spatial discretization phase. Many aspects of the DG methods are investigated and guidelines to choose suitable DG methods are given.

Derived from spatial discretization phase, the semi-discretized system of ODEs consist of three terms of which properties are completely different. The terms corresponding to diffusion and reaction parts are smooth operators but stiff. Their stiffness requires implicit methods for solving. The explicit methods in this phase are simply unpractical because if they are used, time step size must be very small to ensure stability of numerical solution and it makes computational cost unacceptable. The term obtained from advection part contains in itself non-smooth operators such as flux limiter, max, min functions. Non-stiffness of this discretized operator is fit for explicit temporal integration methods. Moreover the explicit methods are physically well-suited because speed of transporting information caused by advection operator is just finite. We propose in this thesis a whole new class of methods that comprise implicit methods for discretized diffusion reaction terms and explicit ones for discretized advection terms. More careful treatments with discretized advection terms are also offered in order to preserve positivity of numerical solutions.

1.2 Outline

The remainder of this thesis is separated into three chapters. Full discretization procedure for ADR equations is given in Chapters 2 and 3. Some numerical results of practical models are provided in Chapter 4.

Chapter 2 deals with the spatial discretization phase. It introduces several DG methods for advection and diffusion-reaction terms. These methods are carefully analyzed in many aspects in order to give out a guideline for choosing DG methods to fit individual problems with specific demands.

Chapter 3 introduces a new temporal integrations methods called Rosenbrock explicit Runge Kutta methods and their variants called Rosenbrock strong-stability preserving methods. These methods are implicit for discretized diffusion-reaction terms and explicit for discretized advection one.

Chapter 4 demonstrates the full discretization process established in previous chapters via numerical results of two practical models. One model concerns evolution of bacteria and the other describes development of tumor-induced blood vessels.

2

Spatial discretization using discontinuous Galerkin methods

Spatial discretization is the first phase of the method of lines for solving PDEs. Among many spatially discretizing methods such as finite difference, finite volume, and finite element methods, discontinuous Galerkin (DG) methods using high order approximation have become an attractive one for the solutions of advection-diffusion-reaction equations. In this chapter we investigate the DG methods applying to the hyperbolic and parabolic problems.

2.1 The conservation laws

Proposed by Chavent and Salzano [10] and then developed by several authors [9, 11–14], DG methods at first was aimed as a high resolution solver for hyperbolic equations.

We consider the initial-boundary value problem associated with the hyperbolic conservation law

$$\frac{\partial u}{\partial t} + \operatorname{div} f(u) = 0 \quad \text{in } (0, T) \times \Omega, \tag{2.1.1a}$$

where $\Omega \subset \mathbb{R}^d$, $u = (u_1, \ldots, u_m)^T$, and f satisfies that any real combination of Jacobian matrices

$$\sum_{j=1}^{d} \xi_j \frac{\partial f_j}{\partial u}$$

has m real eigenvalues and a complete set of eigenvectors. The initial value and boundary condition are given as

$$u(0, x) = u_0 \text{ in } \Omega, \qquad u_0 \in L^\infty(\Omega), \tag{2.1.1b}$$

$$u(t, x) = \gamma \text{ in } (0, T) \times \partial\Omega, \qquad \gamma \in L^\infty((0, T) \times \partial\Omega). \tag{2.1.1c}$$

The simplest example for this type of equations is a linear advection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \qquad a > 0.$$

This equation models the advection of a tracer along with the fluid. The general solution of this equation is very easy to determine. Any smooth function of the form $u(x, t) = u_0(x - at)$ satisfies the above linear advection equation.

2.1.1 Semi-discretized system

To spatially discretize the (2.1.1), we proceed through the following phases:

i. Define a space of test functions in which the exact solution of (2.1.1) is approximated;

ii. Rewrite the original problem (2.1.1) as a finite dimensional version. It results a system of ordinary differential equations (ODEs) for which the integration methods are introduced in the next chapter.

First, we introduce a space of test functions. The domain Ω is approximated by nE non-overlapping elements D^k. This division is called the triangulation of Ω and denoted by $\mathcal{T}_h$. The space of test functions is defined as

$$V_h = \left\{ v_h \in L^2(\Omega) \,|\, v_h|_{D^k} \in P_p\left(D^k\right), \forall D^k \in \mathcal{T}_h \right\}, \tag{2.1.2}$$

where $P_p\left(D^k\right)$ is the space of polynomial functions of degree at most $p \geqslant 1$ on D^k. Note that the space V_h is a *broken* one, i. e. a test function $v_h \in V_h$ is not required to be

continuous at the boundary of a local element $D^k \in \mathcal{T}_h$. It means that the approximate solution u_h of u on V_h possesses more degree of freedoms (DOFs) on boundary of each local element D^k in compare to classical finite element methods in which test functions are continuous between local elements. These extra DOFs allow DG methods to capture correctly wave propagations in the hyperbolic problem (2.1.1). This is the major advantage of the DG methods over classical finite element methods. It is easy to see that the drawback of this approach cause that DG methods consume more computer memory to store the additional DOFs than that of classical finite element methods do.

To obtain approximate solution u_h of (2.1.1), the DG methods employ the Galerkin's idea that is projecting the problem onto the finite dimensional test function space V_h. To do that we multiply (2.1.1a) by $v_h \in V_h$, integrate over $D^k \in \mathcal{T}_h$, and replace the exact solution u by its approximation $u_h \in V_h$:

$$\frac{d}{dt}\int_{D^k} u_h(t,x)\, v_h(x)\, dx + \int_{D^k} \operatorname{div} f(u_h(t,x))\, v_h(x)\, dx = 0, \qquad \forall v_h \in V_h, \forall D^k \in \mathcal{T}_h.$$

Integrating by parts formally, we obtain

$$\frac{d}{dt}\int_{D^k} u_h(t,x)\, v_h(x)\, dx + \sum_{\partial D_e^k \in \partial D^k} \int_{\partial D_e^k} f(u_h(t,x)) \cdot \vec{n}|_{\partial D_e^k} v_h(x)\, d\Gamma - \int_{D^k} f(u_h(t,x)) \cdot \nabla v_h(x)\, dx = 0, \qquad \forall v_h \in V_h, \forall D^k \in \mathcal{T}_h,$$

in which ∂D_e^k is an edge of the boundary of element D^k, $\vec{n}|_{\partial D_e^k}$ is the outward unit normal vector of the element D^k to the edge ∂D_e^k. It should notice that because of the lack continuity of the local solution and the test functions, the quantity $f(u_h(t,x))|_{\partial D_e^k}$ is multiply defined. Therefore we need to choose which solution or combination of solutions is reasonable. This choice is known as *numerical flux* and denoted by $\widehat{f}(u_h)$. Now, let us define problems of finding the numerical solution u_h.

Definition 2.1.1. *The semi-discretized system corresponding to the conservation law (2.1.1) is given as: Find $u_h \in V_h$ such that for all test functions $v_h \in V_h$ and for all element $D^k \in \mathcal{T}_h$,*

we have

$$\frac{d}{dt}\int_{D^k} u_h(t,x)\, v_h(x)\, dx + \sum_{\partial D_e^k \in \partial D^k} \int_{\partial D_e^k} \widehat{f}(u_h(t,x)) \cdot \vec{n}|_{\partial D_e^k} v_h(x)\, d\Gamma - \int_{D^k} f(u_h(t,x)) \cdot \nabla v_h(x)\, dx = 0, \quad (2.1.3)$$

where the numerical flux $\widehat{f}$ is determined later.

Hence, only thing left to fulfill the semi-discretized system (2.1.3) is to determine the numerical flux $\widehat{f}$.

2.1.2 Numerical flux

Before specifying numerical flux $\widehat{f}$, we give out some notations. For each pair element-edge $\left(D^k, \partial D_e^k\right)$, there is a corresponding pair $\left(D^{\tilde{k}}, \partial D_{\tilde{e}}^{\tilde{k}}\right)$ in which two elements D^k, $D^{\tilde{k}}$ are adjacent ones sharing the common edge $\partial D_e^k \equiv \partial D_{\tilde{e}}^{\tilde{k}}$. We can write explicitly two arguments that the numerical flux $\widehat{f}$ defined on edge ∂D_e^k depends on

$$\widehat{f}|_{\partial D_e^k} \equiv \widehat{f}\left(u_h|_{\partial D_e^k}, u_h|_{\partial D_{\tilde{e}}^{\tilde{k}}}\right).$$

For simplicity, we restrict ourselves in this section to consider the scalar case of (2.1.1), i. e. $m = 1$.

The numerical flux $\widehat{f}$ is chosen such that our schemes are alike monotone schemes because monotone schemes which although only first order of accuracy are very stable and converge to the entropy solution. More precisely, we want that in the case that the approximate solutions u_h are element-wise constant function, i.e. $p = 0$, our semi-discretization gives a monotone scheme. Therefore, numerical flux $\widehat{f}$ must satisfy following conditions:

i. $\widehat{f}$ is locally Lipschitz continuous and consistent with the flux f, i.e., $\widehat{f}(u,u) = f(u)$;

ii. $\widehat{f}(\cdot,\cdot)$ is a non-decreasing function of its first argument and a non-increasing function of its second argument.

There are some examples of numerical fluxes satisfying the above requirements in one dimensional case.

- Godunov flux:

$$\widehat{f}(a,b) = \begin{cases} \min_{a\leqslant u\leqslant b} f(u) & \text{if } a \leqslant b, \\ \max_{b\leqslant u\leqslant a} f(u) & \text{if } a > b; \end{cases} \tag{2.1.4a}$$

- Engquist-Osher flux:

$$\widehat{f}(a,b) = \int_0^b \min\left\{f'(u), 0\right\} du + \int_0^a \max\left\{f'(u), 0\right\} du + f(0); \tag{2.1.4b}$$

- Local Lax-Friedrichs flux (also known as Rusanov flux):

$$\widehat{f}(a,b) = \frac{1}{2}\left[f(a) + f(b) + C^{\mathrm{lLF}}(a-b)\right], \tag{2.1.4c}$$

with

$$C^{\mathrm{lLF}} = \max_{u\in \mathrm{I}(a,b)} \left|f'(s)\right|, \quad \mathrm{I}(a,b) = \left[\min\{a,b\}, \max\{a,b\}\right];$$

- Lax-Friedrichs flux:

$$\widehat{f}(a,b) = \frac{1}{2}\left[f(a) + f(b) + C^{\mathrm{LF}}(a-b)\right], \tag{2.1.4d}$$

with

$$C^{\mathrm{LF}} = \max_{\inf u_0 \leqslant u \leqslant \sup u_0} \left|f'(u)\right|;$$

- Roe flux with entropy fix:

$$\widehat{f}(a,b) = \begin{cases} f(a), & \text{if } f'(u) \geqslant 0 \text{ for } u \in \mathrm{I}(a,b), \\ f(b), & \text{if } f'(u) \leqslant 0 \text{ for } u \in \mathrm{I}(a,b), \\ \text{as in (2.1.4c)}, & \text{otherwise.} \end{cases} \tag{2.1.4e}$$

We also give out some examples of numerical fluxes in two dimensional case with $f = [f^x, f^y]^T$ and $\widehat{f} = \left[\widehat{f^x}, \widehat{f^y}\right]$:

- Godunov flux:

$$\widehat{f^x}(a,b) = \begin{cases} \min\limits_{a\leqslant u\leqslant b} f^x(u) & \text{if } a \leqslant b, \\ \max\limits_{b\leqslant u\leqslant a} f^x(u) & \text{if } a > b, \end{cases} \quad \widehat{f^y}(a,b) = \begin{cases} \min\limits_{a\leqslant u\leqslant b} f^y(u) & \text{if } a \leqslant b, \\ \max\limits_{b\leqslant u\leqslant a} f^y(u) & \text{if } a > b. \end{cases} \tag{2.1.5a}$$

- Local Lax-Friedrichs flux:

$$\widehat{f}(a,b) = \frac{1}{2}\left[f(a) + f(b) + C^{\mathrm{lLF}}\vec{n}(a-b)\right], \tag{2.1.5b}$$

with

$$C^{\mathrm{lLF}} = \max_{u\in \mathrm{I}(a,b)} \left|\vec{n}\cdot f'\right|.$$

2.1.3 Weak formulation

After choosing a right numerical flux, the integral on D^k and the path one on ∂D_e^k in (2.1.3) have to be approximated by quadrature rules as follows

$$\int_{D^k} f(u_h(t,x))\cdot\nabla v_h(x)\,dx \approx \sum_{j=1}^{\mathrm{nP}} \omega_j f\left(u_h\left(t,x_j^k\right)\right)\cdot\nabla v_h\left(x_j^k\right)\left|D^k\right|, \tag{2.1.6a}$$

$$\int_{\partial D_e^k} \widehat{f}(u_h(t,x))\cdot\vec{n}|_{\partial D_e^k} v_h(x)\,d\Gamma \approx \sum_{l=1}^{\mathrm{nN}} \theta_l \widehat{f}\left(u_h\left(t,x_{e,l}^k\right)\right)\cdot\vec{n}|_{x_{e,l}^k} v_h\left(x_{e,l}^k\right)\left|\partial D_e^k\right|. \tag{2.1.6b}$$

Here, $\left\{\omega_j, x_j^k\right\}_{j=1,\ldots,\mathrm{nP}}$ is quadrature rule on element D^k and $\left\{\theta_l, x_{e,l}^k\right\}_{l=1,\ldots,\mathrm{nN}}$ is quadrature rule on edge ∂D_e^k. Finally we arrive at the last expression of the semi-discretized

equations

$$\frac{d}{dt}\int_{D^k} u_h(t,x)\, v_h(x)\, dx + \sum_{j=1}^{\mathrm{nP}} \omega_j f\left(u_h\left(t, x_j^k\right)\right)\cdot \nabla v_h\left(x_j^k\right)\left|D^k\right|$$

$$- \sum_{\partial D_e^k \in \partial D^k} \sum_{l=1}^{\mathrm{nN}} \theta_{lf}\widehat{f}\left(u_h\left(t, x_{e,l}^k\right)\right)\cdot \vec{n}|_{x_{e,l}^k} v_h\left(x_{e,l}^k\right)\left|\partial D_e^k\right| = 0, \qquad \forall v_h \in V_h,\ \forall D^k \in \mathcal{T}_h. \tag{2.1.7}$$

We call (2.1.7) weak formulation corresponding to (2.1.1). These equations can be rewritten in ordinary differential equation form as

$$\frac{du_h}{dt} = L_h(u_h),$$

where $L_h : V_h \to V_h$ and for all $v_h \in V_h$,

$$\frac{d}{dt}\int_\Omega u_h v_h dx = \int_\Omega L_h(u_h)\, v_h dx.$$

The operator $L_h(u_h)$ can be seen as a discretized approximation of $\mathrm{div} f(u)$ together with boundary conditions. The following theorem [11] gives an estimation of the quality of this approximation.

Theorem 2.1.1. *Let $f(u) \in W^{p+2,\infty}(\Omega)$. Let the quadrature rule over the edges be exact for polynomials of degree $2p+1$, and let the one over the element be exact for polynomials of degree $2p$. Assume that the triangulation $\mathcal{T}_h$ is regular, i.e., that there is a constant σ such that*

$$\frac{h_{D^k}}{\rho_{D^k}} \geqslant \sigma, \qquad \forall D^k \in \mathcal{T}_h,$$

where h_{D^k} is the diameter of D^k, and ρ_{D^k} is the diameter of the biggest ball included in D^k. Then,

$$\|L_h(u) + \mathrm{div} f(u)\|_{L^\infty(\Omega)} \leqslant C h^{p+1} |f(u)|_{W^{p+2,\infty}(\Omega)}.$$

2.1.4 Numerical examples

Linear advection problem

The very first problem we use to test the DG methods is the simple linear wave equation

$$\frac{\partial u}{\partial t} + 2\pi \frac{\partial u}{\partial x} = 0, \qquad x \in [0, 2\pi],$$

with the initial value $u(x, 0) = \sin(x)$ and the boundary condition is given the left end of the interval $u(0, t) = -\sin(2\pi t)$. It is easy to see that the exact solution is

$$u(x, t) = \sin(x - 2\pi t).$$

The linear advection is spatially discretized with several element sizes

$$h \in \left\{ \frac{2\pi}{2^4}, \frac{2\pi}{2^5}, \frac{2\pi}{2^6}, \frac{2\pi}{2^7} \right\}.$$

The degree of approximate polynomial on local elements is taken from 1 to 3. Numerical flux is chosen as

$$\widehat{2\pi u_h} = \frac{2\pi}{2}\left(u_h^- + u_h^+\right) + \frac{1-a}{2} 2\pi \vec{n}\left(u_h^- - u_h^+\right),$$

in which u^-, u^+ are respectively internal and external values of u_h, $\vec{n}$ is outward normal vector in one dimensional case. Parameter a varies from 0 to 1 in which if $a = 0$ the numerical flux is of upwind and if $a = 1$ the numerical flux is the central one. The upwind numerical flux is chosen for numerical computation.

The numerical results given in Table 2.1.1 practically confirm the order of approximation by DG methods which is $p + 1$ if the degree of approximate polynomial is at most p.

	$p=1$		$p=2$		$p=3$	
h	L^∞ error	Order	L^∞ error	Order	L^∞ error	Order
$2\pi/2^4$	0.0243		0.9912×10^{-3}		0.2783×10^{-4}	
$2\pi/2^5$	0.0063	1.9578	0.1256×10^{-3}	2.9802	0.0176×10^{-4}	3.9814
$2\pi/2^6$	0.0016	1.9801	0.0158×10^{-3}	2.9952	0.0011×10^{-4}	3.9953
$2\pi/2^7$	0.0004	1.9904	0.0020×10^{-3}	2.9959	0.0001×10^{-4}	3.9987

TABLE 2.1.1: *Errors and accurate orders of the numerical solution at* $t = 1.0$.

BURGERS' EQUATION

The second example is inviscid Burgers' equation with periodic boundary condition

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial u^2}{\partial x} = 0, \quad x \in [0,1], t \in [0, 0.05],$$

with the smooth initial value

$$u(0,x) = \frac{1}{4} + \frac{1}{2}\sin(\pi(2x-1)).$$

The element sizes for spatial discretization are

$$h \in \left\{2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}\right\}$$

and the degrees of local approximate polynomials varies from 1 to 3. Because the flux function is nonlinear, we choose the numerical flux as local Lax-Friedrichs one (2.1.4c). In Table 2.1.2 the approximation of DG methods for the nonlinear problem are proved via linear, quadratic, and cubic elements.

	$p=1$		$p=2$		$p=3$	
h	L^2 error	Order	L^2 error	Order	L^2 error	Order
2^{-4}	0.0147		0.9397×10^{-3}		0.5829×10^{-4}	
2^{-5}	0.0045	1.7061	0.1813×10^{-3}	2.3734	0.0532×10^{-4}	3.4541
2^{-6}	0.0014	1.6472	0.0340×10^{-3}	2.4133	0.0042×10^{-4}	3.6488
2^{-7}	0.0005	1.6229	0.0061×10^{-3}	2.4739	0.0003×10^{-4}	3.6028

TABLE 2.1.2: *Estimated errors and accurate orders for solving inviscid Burgers' equation.*

Maxwell's equations

The last numerical demonstration in this section is the two dimensional Maxwell's equations. Its normalized system of equations on domain $\Omega = [-1, 1]^2$ is of the form

$$\begin{aligned}
\frac{\partial H^x}{\partial t} &= -\frac{\partial E^z}{\partial y}, \\
\frac{\partial H^y}{\partial t} &= \frac{\partial E^z}{\partial x}, \\
\frac{\partial E^z}{\partial t} &= \frac{\partial H^y}{\partial x} - \frac{\partial H^x}{\partial y},
\end{aligned}$$

with initial values

$$H^x(0, x, y) = H^y(0, x, y) = 0 \quad \text{and} \quad E^z(0, x, y) = \sin(\pi x)\sin(\pi y).$$

And the boundary condition is $E^z = 0$ on $\partial\Omega$. The exact solution is given as

$$\begin{aligned}
H^x(t, x, y) &= -\frac{1}{\sqrt{2}}\sin(\pi x)\cos(\pi y)\sin\left(\sqrt{2}\pi t\right), \\
H^y(t, x, y) &= \frac{1}{\sqrt{2}}\cos(\pi x)\sin(\pi y)\sin\left(\sqrt{2}\pi t\right), \\
E^z(t, x, y) &= \sin(\pi x)\sin(\pi y)\cos\left(\sqrt{2}\pi t\right).
\end{aligned}$$

We triangulate the domain Ω by unstructured triangles in which sizes of triangle are in the range

$$h \in \left\{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}\right\}.$$

The degrees of approximate polynomial are the same as the two above examples and numerical fluxes are of upwind type. Observing the results shown in Table 2.1.3, we see that the scheme possesses optimal convergence rate, i.e. $\mathcal{O}\left(h^{p+1}\right)$.

2.2 Parabolic equations

In this section we extend DG methods from first order spatial derivatives cases to higher order spatial problems. The idea employing the DG methods for the problems with higher spatial derivatives is rewriting the high spatial derivative as a system of first order equations and then discretize this system. For example, if we need to solve a problem in

	$p=1$		$p=2$		$p=3$	
h	L^2 error	Order	L^2 error	Order	L^2 error	Order
2^{-2}	1.0811×10^{-1}		1.4088×10^{-2}		1.2078×10^{-3}	
2^{-3}	2.2705×10^{-2}	2.2515	1.6785×10^{-3}	3.0692	8.2581×10^{-5}	3.8704
2^{-4}	5.3955×10^{-3}	2.0731	2.0892×10^{-4}	3.0062	4.9124×10^{-6}	4.0713
2^{-5}	1.3150×10^{-3}	2.0367	2.6035×10^{-5}	3.0044	3.0374×10^{-7}	4.0156

TABLE 2.1.3: *Estimated errors and accurate orders for solving normalized Maxwell's equations at* $t = 0.5$.

one dimensional case

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2}, \qquad a > 0,$$

subject to suitable boundary conditions, we apply DG methods to the equivalent system of first order spatial differential equations

$$\begin{aligned} \frac{\partial u}{\partial t} &= a\frac{\partial q}{\partial x}, \\ q &= \frac{\partial u}{\partial x}. \end{aligned}$$

The major disadvantage of discretizing parabolic equations in this manner is to increase the size of the problems and therefore more computational cost. But on the other side, this approach gives us accurate approximation of derivatives used in another parts of ADR equations and provide an unified spatial discretization for ADR equations.

2.2.1 SEMI-DISCRETIZED SYSTEM

For the sake of simplicity we restrict ourselves to the model problem

$$\frac{\partial u}{\partial t} = a\Delta u, \qquad a > 0, \qquad \text{in} \quad \Omega, \tag{2.2.1a}$$

$$u = g_D \qquad \text{on} \quad \partial\Omega_D, \tag{2.2.1b}$$

$$\frac{\partial u}{\partial \vec{n}} = g_N \qquad \text{on} \quad \partial\Omega_N, \tag{2.2.1c}$$

where Ω is a bounded domain in $\mathbb{R}^d$ with boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, $\vec{n}$ is the outward unit normal vector to the boundary of Ω. As mentioned above, to apply DG methods

we rewrite the problem (2.2.1) as a first order system of equations

$$\frac{\partial u}{\partial t} = a\nabla \cdot q \qquad \text{in} \quad \Omega, \tag{2.2.2a}$$

$$q = \nabla u \qquad \text{in} \quad \Omega, \tag{2.2.2b}$$

$$u = g_D \qquad \text{on} \quad \partial\Omega_D, \tag{2.2.2c}$$

$$q \cdot \vec{n} = g_N \qquad \text{on} \quad \partial\Omega_N. \tag{2.2.2d}$$

Next, broken spaces V_h and Σ_h associated with the triangulation $\mathcal{T}_h$ are introduced:

$$V_h = \left\{ v_h \in L^2(\Omega) \,|\, v_h|_{D^k} \in P_p\left(D^k\right), \forall D^k \in \mathcal{T}_h \right\}, \tag{2.2.3}$$

$$\Sigma_h = \left\{ \sigma_h \in [L^2(\Omega)]^d \,|\, v_h|_{D^k} \in \left[P_p\left(D^k\right)\right]^d, \forall D^k \in \mathcal{T}_h \right\}. \tag{2.2.4}$$

Following the discretization process for advection equations, the DG formulations for the (2.2.2) are of the form: Find $u_h \in V_h$ and $q_h \in \Sigma_h$ such that for all elements $D^k \in \mathcal{T}_h$, for all test functions $v_h \in V_h$ and $\sigma_h \in \Sigma_h$, we have

$$\frac{d}{dt}\int_{D^k} u_h v_h dx = -a\int_{D^k} q_h \cdot \nabla v_h dx + a\int_{\partial D^k} \widehat{q}_h \cdot \vec{n} v_h d\Gamma, \tag{2.2.5a}$$

$$\int_{\partial D^k} q_h \cdot \sigma_h dx = -\int_{D^k} u_h \left(\nabla \cdot \sigma_h\right) dx + \int_{\partial D^k} \widehat{u}_h \vec{n} \cdot \sigma_h d\Gamma. \tag{2.2.5b}$$

Here, the numerical fluxes $\widehat{q}_h$ and $\widehat{u}_h$ are approximations to q_h and u_h on boundary of element D^k. They are defined in the next section to complete the DG formulation. For expressing q_h solely on u_h, we prefer to formally take integration by parts. Then, equation (2.2.5b) becomes:

$$\int_{D^k} q_h \cdot \sigma_h dx = \int_{D^k} \nabla u_h \cdot \sigma_h dx + \int_{\partial D^k} \left(\widehat{u}_h - u_h\right) \vec{n} \cdot \sigma_h ds, \qquad \forall \sigma_h \in \Sigma_h. \tag{2.2.5c}$$

Equation (2.2.5c) is going to be used to create the so called primal formulation in the next section.

2.2.2 The primal formulation

In this section, we introduce another form of (2.2.5) called *primal formulation*. In order to do that, we require some additional notations. For each element-edge pair $\left\{ D^k, \partial D_e^k \right\}$

with $\partial D_e^k \not\subset \partial\Omega$, there is a corresponding pair $\left\{D^{\tilde{k}}, \partial D_{\tilde{e}}^{\tilde{k}}\right\}$ in which two elements D^k and $D^{\tilde{k}}$ are adjacent via the edge $\partial D_e^k \equiv \partial D_{\tilde{e}}^{\tilde{k}}$. We denote that the value of u_h and q_h on edges ∂D_e^k and $\partial D_{\tilde{e}}^{\tilde{k}}$ are respectively $u_{h,e}^k$, $q_{h,e}^k$ and $u_{h,\tilde{e}}^{\tilde{k}}$, $q_{h,\tilde{e}}^{\tilde{k}}$. Similarly, the outward unit normal vectors to ∂D_e^k and $\partial D_{\tilde{e}}^{\tilde{k}}$ are denoted respectively by $\vec{n}_e^k$ and $\vec{n}_{\tilde{e}}^{\tilde{k}}$. Using these notations, we define average and jumping operators at the interior edge $\partial D_e^k \equiv \partial D_{\tilde{e}}^{\tilde{k}}$ as follows

$$\{\!\{u_h\}\!\} = \frac{u_{h,e}^k + u_{h,\tilde{e}}^{\tilde{k}}}{2}, \qquad \{\!\{q_h\}\!\} = \frac{q_{h,e}^k + q_{h,\tilde{e}}^{\tilde{k}}}{2},$$
$$[\![u_h]\!] = u_{h,e}^k \vec{n}_e^k + u_{h,\tilde{e}}^{\tilde{k}} \vec{n}_{\tilde{e}}^{\tilde{k}}, \qquad [\![q_h]\!] = q_{h,e}^k \cdot \vec{n}_e^k + q_{h,\tilde{e}}^{\tilde{k}} \cdot \vec{n}_{\tilde{e}}^{\tilde{k}}.$$

Notice that the jump $[\![u_h]\!]$ of the scalar quantity u_h is a vector parallel to the normal, and the jump $[\![q_h]\!]$ of the vector quantity q_h is scalar. These operators is going to be used in definitions of numerical fluxes shown later.

With these notations we can set up formulation of DG methods over the whole domain Ω. Summing up (2.2.5) over all elements D^k of the triangulation $\mathcal{T}_h$, we have: for all $v_h \in V_h$ and for all $\sigma_h \in \Sigma_h$,

$$\frac{d}{dt}\int_\Omega u_h v_h dx = -a\int_\Omega q_h \cdot \nabla_h v_h dx + a\sum_{k=1}^{\mathrm{nE}} \int_{\partial D^k} \vec{n} \cdot \widehat{q}_h v_h d\Gamma, \tag{2.2.6a}$$

$$\int_\Omega q_h \cdot \sigma_h dx = \int_\Omega \nabla_h u_h \cdot \sigma_h dx + \sum_{k=1}^{\mathrm{nE}} \int_{\partial D_e^k} (\widehat{u}_h - u_h)\, \vec{n} \cdot \sigma_h d\Gamma, \tag{2.2.6b}$$

where $\nabla_h v_h$ and $\nabla_h \cdot \sigma_h$ are element-wise operators defined as follows

$$\nabla_h v_h|_{D^k} = \nabla v_h, \qquad \nabla_h \cdot \sigma_h|_{D^k} = \nabla \cdot \sigma_h.$$

To simplify sums over boundary of elements D^k, we employ the jump and average operators with the help of the following lemma.

Lemma 2.2.1. *For any $w_h \in V_h$ and $\tau_h \in \Sigma_h$, we have*

$$\sum_{k=1}^{\mathrm{nE}} \int_{\partial D^k} w_h \vec{n} \cdot \tau_h d\Gamma = \int_{\partial \mathcal{T}_h^\circ} \left([\![w_h]\!] \cdot \{\!\{\tau_h\}\!\} + \{\!\{w_h\}\!\}\, [\![\tau_h]\!]\right) d\Gamma + \int_{\partial\Omega} w_h \vec{n} \cdot \tau_h d\Gamma, \tag{2.2.7}$$

where

$$\partial\mathcal{T}_h^\circ = \bigcup_{n=1}^{\mathrm{nE}} \partial D^k \setminus \partial\Omega.$$

Proof. We have

$$\sum_{k=1}^{\mathrm{nE}} \int_{\partial D^k} w_h\tau_h \cdot \vec{n} d\Gamma = \sum_{k=1}^{\mathrm{nE}} \sum_{\partial D_e^k \in \partial D^k} \int_{\partial D_e^k \in \partial D^k} w_h\tau_h \cdot \vec{n} d\Gamma.$$

For each internal edge ∂D_e^k, there is only one adjacent edge $\partial D_{\tilde{e}}^{\tilde{k}}$ such that two element D^k are $D^{\tilde{k}}$ are adjacent via edge ∂D_e^k and $\partial D_{\tilde{e}}^{\tilde{k}}$. The sum over these edge is

$$\begin{aligned}\int_{\partial D_e^k} w_h\tau_h \cdot \vec{n} d\Gamma + \int_{\partial D_{\tilde{e}}^{\tilde{k}}} w_h\tau_h \cdot \vec{n} d\Gamma \\ = \int_{\partial D_e^k \cup \partial D_{\tilde{e}}^{\tilde{k}}} \left(w_{h,e}^k \tau_{h,e}^k \cdot \vec{n}_{h,e}^k + w_{h,\tilde{e}}^{\tilde{k}} \tau_{h,\tilde{e}}^{\tilde{k}} \cdot \vec{n}_{h,\tilde{e}}^{\tilde{k}} \right) d\Gamma.\end{aligned}$$

The integrand of the right hand side integral is rewritten as

$$\begin{aligned} & w_{h,e}^k \tau_{h,e}^k \cdot \vec{n}_{h,e}^k + w_{h,\tilde{e}}^{\tilde{k}} \tau_{h,\tilde{e}}^{\tilde{k}} \cdot \vec{n}_{h,\tilde{e}}^{\tilde{k}} \\ & \quad = \frac{1}{2} w_{h,e}^k \left(\tau_{h,e}^k + \tau_{h,\tilde{e}}^{\tilde{k}} \right) \cdot \vec{n}_e^k + \frac{1}{2} \left(w_{h,e}^k + w_{h,\tilde{e}}^{\tilde{k}} \right) \tau_{h,e}^k \cdot \vec{n}_e^k \\ & \qquad + \frac{1}{2} w_{h,\tilde{e}}^{\tilde{k}} \left(\tau_{h,\tilde{e}}^{\tilde{k}} + \tau_{h,e}^k \right) \cdot \vec{n}_{\tilde{e}}^{\tilde{k}} + \frac{1}{2} \left(w_{h,e}^k + w_{h,\tilde{e}}^{\tilde{k}} \right) \tau_{h,\tilde{e}}^{\tilde{k}} \cdot \vec{n}_{\tilde{e}}^{\tilde{k}} \\ & \qquad - \frac{1}{2} w_{h,e}^k \tau_{h,\tilde{e}}^{\tilde{k}} \cdot \vec{n}_e^k - \frac{1}{2} w_{h,\tilde{e}}^{\tilde{k}} \tau_{h,e}^k \cdot \vec{n}_e^k \\ & \qquad - \frac{1}{2} w_{h,\tilde{e}}^{\tilde{k}} \tau_{h,e}^k \cdot \vec{n}_{\tilde{e}}^{\tilde{k}} - \frac{1}{2} w_{h,e}^k \tau_{h,\tilde{e}}^{\tilde{k}} \cdot \vec{n}_{\tilde{e}}^{\tilde{k}} \\ & \quad = [\![w_h]\!] \cdot \{\!\{ \tau_h \}\!\} + \{\!\{ w_h \}\!\} [\![\tau_h]\!]. \end{aligned}$$

The last equality is obtained by noticing that $\vec{n}_e^k = -\vec{n}_{\tilde{e}}^{\tilde{k}}$. □

Now, we present q_h solely depending on u_h. Applying (2.2.7) to (2.2.6b), we have

$$\begin{aligned} \int_\Omega q_h \cdot \sigma_h dx = & \int_\Omega \nabla_h u_h \cdot \sigma_h dx \\ & + \int_{\partial\mathcal{T}_h} \left([\![\widehat{u}_h - u_h]\!] \cdot \{\!\{ \sigma_h \}\!\} + \{\!\{ \widehat{u}_h - u_h \}\!\} [\![\sigma_h]\!] \right) d\Gamma \qquad (2.2.8) \\ & + \int_{\partial\Omega} \left(\widehat{u}_h - u_h \right) \vec{n} \cdot \sigma_h d\Gamma. \end{aligned}$$

Defining lifting operators $r : \left[L^2\left(\partial\mathcal{T}_h^\circ\right)\right]^d \to \Sigma_h$, $l : L^2\left(\partial\mathcal{T}_h^\circ\right) \to \Sigma_h$, and $r_D : L^2(\partial\Omega_D) \to \Sigma_h$ by

$$\begin{aligned}
\int_\Omega r(\tau)\cdot\sigma_h dx &= -\int_{\partial\mathcal{T}_h^\circ} \tau\cdot\{\!\{\sigma_h\}\!\}\, d\Gamma,\\
\int_\Omega l(w)\cdot\sigma_h dx &= -\int_{\partial\mathcal{T}_h^\circ} w\,[\![\sigma_h]\!]\, d\Gamma,\\
\int_\Omega r_D(w)\cdot\sigma_h dx &= -\int_{\partial\Omega_D} w\vec{n}\cdot\sigma d\Gamma,
\end{aligned} \tag{2.2.9}$$

for all $\sigma_h \in \Sigma_h$, respectively, (2.2.9) can be rewritten as

$$q_h = \nabla_h u_h - r\left([\![\widehat{u}_h - u_h]\!]\right) - l\left(\{\!\{\widehat{u}_h - u_h\}\!\}\right) - r_D\left(\widehat{u}_h - u_h\right). \tag{2.2.10}$$

Bringing this expression of q_h into (2.2.6a) after applying (2.2.7) for this formula, we finally arrive at the primal formulation of DG methods

$$\frac{d}{dt}\int_\Omega u_h v_h dx = aA_h\left(u_h, v_h\right), \quad \forall v_h \in V_h, \tag{2.2.11}$$

where

$$\begin{aligned}
A_h\left(u_h, v_h\right) := &-\int_\Omega \nabla_h u_h\cdot\nabla_h v_h dx\\
&+\int_{\partial\mathcal{T}_h^\circ}\left(\{\!\{\widehat{q}_h\}\!\}\cdot[\![v_h]\!] - [\![\widehat{u}_h - u_h]\!]\cdot\{\!\{\nabla_h v_h\}\!\}\right) d\Gamma\\
&-\int_{\partial\mathcal{T}_h^\circ}\left(\{\!\{\widehat{u}_h - u_h\}\!\}\,[\![\nabla_h v_h]\!] - [\![\widehat{q}_h]\!]\,\{\!\{v_h\}\!\}\right) d\Gamma\\
&+\int_{\partial\Omega}\widehat{q}_h\cdot\vec{n}v_h d\Gamma - \int_{\partial\Omega_D}\left(g_D - u_h\right)\vec{n}\cdot\nabla_h v_h d\Gamma.
\end{aligned} \tag{2.2.12a}$$

The proposition below gives us the consistency of primal formulation.

Proposition 2.2.1. *If the numerical fluxes are consistent, then the primal formulation is also consistent.*

It means that if u is exact solution of (2.2.1) and

$$\widehat{u}_h(u) = u, \quad \widehat{q}_h(u) = \nabla u, \qquad \text{on } \partial\mathcal{T}_h,$$

then

$$\frac{d}{dt}\int_{\Omega} uv_h dx = A_h\left(u, v_h\right), \qquad \forall v_h \in V_h.$$

Proof. Since u is an exact solution of (2.2.1), on all interior edges of triangulation $\mathcal{T}_h$ we have

$$\left\{\!\left\{\widehat{q}_h\left(u\right)\right\}\!\right\} = \nabla u,\ \left[\!\left[\widehat{u}_h\left(u\right) - u\right]\!\right] = 0,\ \left[\!\left[\widehat{q}_h\left(u\right)\right]\!\right] = 0,\ \left\{\!\left\{\widehat{u}_h\left(u\right) - u\right\}\!\right\} = 0.$$

Therefore

$$A_h\left(u, v_h\right) = -\int_{\Omega} \nabla u \cdot \nabla v_h dx + \int_{\partial \mathcal{T}_h^{\circ}} \nabla u \cdot \left[\!\left[v_h\right]\!\right] d\Gamma + \int_{\partial \Omega_D} \nabla u \cdot \vec{n} v_h d\Gamma + \int_{\partial \Omega_N} g_N v_h d\Gamma.$$

By integration by parts and using the formula (2.2.7), we have for any $v_h \in V_h$

$$\begin{aligned}
\int_{\Omega} \nabla u \cdot \nabla_h v_h dx &= -\int_{\Omega} \Delta u v_h dx + \sum_{k=1}^{\mathrm{nE}} \int_{D^k} \vec{n} \cdot \nabla u v_h d\Gamma \\
&= -\frac{1}{a}\frac{d}{dt}\int_{\Omega} u v_h dx + \int_{\partial \mathcal{T}_h^{\circ}} \left(\left\{\!\left\{\nabla u\right\}\!\right\} \cdot \left[\!\left[v_h\right]\!\right] + \left[\!\left[\nabla u\right]\!\right] \left\{\!\left\{v_h\right\}\!\right\}\right) d\Gamma \\
&\quad + \int_{\partial \Omega} \nabla u \cdot \vec{n} v_h d\Gamma \\
&= -\frac{1}{a}\frac{d}{dt}\int_{\Omega} u v_h dx + \int_{\partial \mathcal{T}_h^{\circ}} \nabla u \cdot \left[\!\left[v_h\right]\!\right] d\Gamma \\
&\quad + \int_{\partial \Omega_D} \nabla u \cdot \vec{n} v_h d\Gamma + \int_{\partial \Omega_N} g_N v_h d\Gamma.
\end{aligned}$$

Then,

$$\frac{d}{dt}\int_{\Omega} u v_h dx = a A_h\left(u, v_h\right).$$

□

Recalling that the numerical fluxes are a single-valued function on each edge of $\partial \mathcal{T}_h$ and utilizing the property of jump operator which vanishes for single-valued inter-element

fluxes, the primal form (2.2.12a) is reduced to

$$\begin{aligned} A_h(u_h, v_h) = &- \int_\Omega \nabla_h u_h \cdot \nabla_h v_h dx \\ &+ \int_{\partial \mathcal{T}_h^\circ} \left([\![u_h]\!] \cdot \{\!\{ \nabla_h v_h \}\!\} - \{\!\{ \widehat{u}_h - u_h \}\!\} [\![\nabla_h v_h]\!] + \widehat{q}_h \cdot [\![v_h]\!] \right) d\Gamma \\ &- \int_{\partial \Omega_D} (\widehat{u}_h - u_h) \vec{n} \cdot \nabla_h v_h d\Gamma + \int_{\partial \Omega} \widehat{q}_h \cdot \vec{n} v_h d\Gamma. \end{aligned} \tag{2.2.12b}$$

2.2.3 Numerical fluxes for parabolic problems

In this part we introduce four numerical fluxes and their formulations in primal form. These numerical fluxes are based on jumping, $[\![\cdot]\!]$, and average, $\{\!\{\cdot\}\!\}$, operators introduced before.

Central method

The problem (2.2.1) is symmetric, i.e., there is no preferred direction of propagation, then it is natural to consider the central flux [6]:

$$\widehat{u}_h = \{\!\{ u_h \}\!\}, \qquad \widehat{q}_h = \{\!\{ q_h \}\!\}, \qquad \text{on } \partial \mathcal{T}_h, \tag{2.2.13a}$$

$$\widehat{u}_h = g_D, \qquad \widehat{q}_h = q_h, \qquad \text{on } \partial \Omega_D, \tag{2.2.13b}$$

$$\widehat{u}_h = u_h, \qquad \widehat{q}_h = g_N \vec{n}, \qquad \text{on } \partial \Omega_N. \tag{2.2.13c}$$

These numerical fluxes give

$$\{\!\{ \widehat{u}_h - u_h \}\!\} = 0, \quad [\![\widehat{q}_h]\!] = 0,$$

and

$$q_h = \nabla_h u_h + r([\![u_h]\!]) - r_D (g_D - u_h).$$

Summands of $A_h(u_h, v_h)$ are computed as follows.

$$\begin{aligned}\int_{\partial\mathcal{T}_h^\circ} \widehat{q}_h \cdot [\![v_h]\!]\, d\Gamma &= -\int_\Omega \left(\nabla_h u_h + r([\![u_h]\!]) - r_D\left(g_D - u_h\right)\right) \cdot r([\![v_h]\!])\, dx \\ &= -\int_\Omega \left(\nabla_h u_h + r([\![u_h]\!]) + r_D(u_h)\right) \cdot r([\![v_h]\!])\, dx \\ &\quad - \int_{\partial\Omega_D} g_D \vec{n} \cdot r([\![v_h]\!])\, d\Gamma,\end{aligned}$$

$$\begin{aligned}\int_{\partial\mathcal{T}_h^\circ} [\![u_h]\!] \cdot \{\!\{\nabla_h v_h\}\!\}\, d\Gamma &= -\int_\Omega r([\![u_h]\!]) \cdot \nabla_h v_h dx, \\ -\int_{\partial\mathcal{T}_h^\circ} \{\!\{\widehat{u}_h - u_h\}\!\}\, [\![\nabla_h v_h]\!]\, d\Gamma &= 0,\end{aligned}$$

and

$$\begin{aligned}\int_{\partial\Omega} \widehat{q}_h \cdot \vec{n} v_h d\Gamma &= \int_{\partial\Omega_D} q_h \cdot \vec{n} v_h d\Gamma + \int_{\partial\Omega_N} g_N v_h d\Gamma \\ &= -\int_\Omega \left(\nabla_h u_h + r([\![u_h]\!]) - r_D\left(g_D - u_h\right)\right) \cdot r_D(v_h)\, dx \\ &\quad + \int_{\partial\Omega_N} g_N \vec{n} v_h d\Gamma.\end{aligned}$$

The primal formulation using the central method is of the form

$$\frac{d}{dt}\int_\Omega u_h v_h dx = a A_h^{\mathtt{Cent}}(u_h, v_h) = a\left(B_h^{\mathtt{Cent}}(u_h, v_h) + C_h^{\mathtt{Cent}}(v_h)\right), \qquad (2.2.14a)$$

in which

$$\begin{aligned}B_h^{\mathtt{Cent}}(u_h, v_h) &= -\int_\Omega \nabla_h u_h \cdot \nabla_h v_h dx \\ &\quad - \int_\Omega \left(\nabla_h u_h \cdot r([\![v_h]\!]) + r([\![u_h]\!]) \cdot \nabla_h v_h\right) dx \\ &\quad - \int_\Omega \left(\nabla_h u_h \cdot r_D(v_h) + r_D(u_h) \cdot \nabla_h v_h\right) dx \\ &\quad - \int_\Omega \left(r([\![u_h]\!]) + r_D(u_h)\right) \cdot \left(r([\![v_h]\!]) + r_D(v_h)\right) dx,\end{aligned} \qquad (2.2.14b)$$

and

$$C_h^{\text{Cent}}(u_h, v_h) = -\int_{\partial\Omega_D} g_D \vec{n} \cdot (\nabla_h v_h + r([\![v_h]\!]) + r_D(v_h))\, d\Gamma + \int_{\partial\Omega_N} g_N v_h d\Gamma. \tag{2.2.14c}$$

INTERIOR PENALTY (IP) METHOD

The second choice for numerical fluxes is interior penalty method [2] devised in the late 1970s. Numerical fluxes of IP method are given as

$$\widehat{u}_h = \{\!\{u_h\}\!\}, \quad \widehat{q}_h = \{\!\{\nabla_h u_h\}\!\} - \mu [\![u_h]\!], \quad \text{on } \partial\mathcal{T}_h^\circ, \tag{2.2.15a}$$

$$\widehat{u}_h = g_D, \quad \widehat{q}_h = \nabla_h u_h - \mu u_h \vec{n}_e^k, \quad \text{on } \partial\Omega_D, \tag{2.2.15b}$$

$$\widehat{u}_h = u_h, \quad \widehat{q}_h = g_N \vec{n} \quad \text{on } \partial\Omega_N. \tag{2.2.15c}$$

Here, μ is the penalty parameter defined on each edge of element D^k. The primal formulation for IP method is obtained in the similar manner of the central method which is

$$\frac{d}{dt}\int_\Omega u_h v_h dx = a A_h^{\text{IP}}(u_h, v_h) = a\left(B_h^{\text{IP}}(u_h, v_h) + C_h^{\text{IP}}(v_h)\right), \tag{2.2.16a}$$

with

$$B_h^{\text{IP}}(u_h, v_h) = -\int_\Omega (\nabla_h u_h \cdot \nabla_h v_h + \nabla_h u_h \cdot r([\![v_h]\!]) + r([\![u_h]\!]) \cdot \nabla_h v_h)\, dx - \int_\Omega (\nabla_h u_h \cdot r_D(v_h) + r_D(u_h) \cdot \nabla_h v_h)\, dx - \int_{\partial\mathcal{T}_h^\circ} \mu [\![u_h]\!] \cdot [\![v_h]\!]\, d\Gamma - \int_{\partial\Omega_D} \mu u_h v_h d\Gamma, \tag{2.2.16b}$$

and

$$C_h^{\text{IP}}(v_h) = -\int_{\partial\Omega_D} g_D \vec{n} \cdot \nabla_h v_h d\Gamma + \int_{\partial\Omega_N} g_N v_h d\Gamma. \tag{2.2.16c}$$

Local DG (LDG) method

The third numerical fluxes is local DG method [15] in which the fluxes $\widehat{u}_h$ and $\widehat{q}_h$ on interior edges are given by

$$\widehat{u}_h = \{\!\{u_h\}\!\} - C_{12} \cdot [\![u_h]\!], \quad \widehat{q}_h = \{\!\{q_h\}\!\} - C_{11} [\![u_h]\!] + C_{12} [\![q_h]\!]. \tag{2.2.17a}$$

On the boundary edges, they are defined as

$$\widehat{u}_h = g_D, \qquad \widehat{q}_h = q_h - C_{11} (u - g_D) \vec{n}, \qquad \text{on } \partial\Omega_D, \tag{2.2.17b}$$

$$\widehat{u}_h = u_h, \qquad \widehat{q}_h = g_N \vec{n}, \qquad \text{on } \partial\Omega_N. \tag{2.2.17c}$$

Here, C_{11} is a positive constant and C_{12} is a vector which is determined for each interior edge.

The primal form for LDG methods is

$$\frac{d}{dt} \int_\Omega u_h v_h dt = a A_h^{\mathrm{LDG}} (u_h, v_h) = a \left(B_h^{\mathrm{LDG}} (u_h, v_h) + C_h^{\mathrm{LDG}} (v_h) \right). \tag{2.2.18a}$$

Here, the operators B_h^{LDG} and C_h^{LDG} are given by

$$\begin{aligned} B_h^{\mathrm{LDG}} (u_h, v_h) - \int_\Omega & \left(\nabla_h u_h + r([\![u_h]\!]) + l(C_{12} \cdot [\![u_h]\!]) + r_D(u_h) \right) \\ & \cdot \left(\nabla_h v_h + r([\![v_h]\!]) + l(C_{12} \cdot [\![v_h]\!]) + r_D(v_h) \right) dx \\ - \int_{\partial \mathcal{T}_h^\circ} & C_{11} [\![u_h]\!] \cdot [\![v_h]\!] \, d\Gamma - \int_{\partial\Omega_D} C_{11} u_h v_h d\Gamma, \end{aligned} \tag{2.2.18b}$$

and

$$\begin{aligned} C_h^{\mathrm{LDG}} (v_h) = & - \int_{\partial\Omega_D} g_D \vec{n} \cdot \left(\nabla_h v_h + r([\![v_h]\!]) + l(C_{12} \cdot [\![v_h]\!]) + r_D(v_h) \right) d\Gamma \\ & + \int_{\partial\Omega_D} C_{11} g_D v_h d\Gamma + \int_{\partial\Omega_N} g_N v_h d\Gamma. \end{aligned} \tag{2.2.18c}$$

Compact DG (CDG) method

The CDG method [29] is similar to LDG one with more careful treatment on inter-element numerical fluxes. The numerical fluxes are given as

$$\widehat{u}_h = \begin{cases} \{\!\{u_h\}\!\} - C_{12} \cdot [\![u_h]\!] & \text{on } \partial\mathcal{T}_h^\circ, \\ g_D & \text{on } \partial\Omega_D, \\ u_h & \text{on } \partial\Omega_N, \end{cases} \tag{2.2.19a}$$

$$\widehat{q}_h = \begin{cases} \{\!\{q_h^e\}\!\} - C_{11}[\![u_h]\!] + C_{12}[\![q_h^e]\!] & \text{on each edge } e \in \partial\mathcal{T}_h^\circ, \\ q_h^e - C_{11}\left(u_h - g_D\right)\vec{n} & \text{on each edge } e \in \partial\Omega_D, \\ g_N\vec{n} & \text{on } \partial\Omega_N. \end{cases} \tag{2.2.19b}$$

Here, for each edge $e \in \partial\mathcal{T}_h$, q_h^e is defined by: for all $\sigma_h \in \Sigma_h$, for all elements $D^k \in \mathcal{T}_h$,

$$\int_{D^k} q_h^e \cdot \sigma_h dx = \int_{D^k} \nabla_h u_h \cdot \sigma_h dx + \int_{\partial D^k} (\widehat{u}_h^e - u_h)\,\vec{n} \cdot \sigma_h d\Gamma, \tag{2.2.19c}$$

with

$$\widehat{u}_h^e = \begin{cases} \widehat{u}_h & \text{on edge } e, \\ u_h & \text{otherwise.} \end{cases} \tag{2.2.19d}$$

Instead of using q_h in the numerical fluxes, CDG introduces a new quantity q_h^e which can be seen as a refinement of q_h on each edge e of $\partial\mathcal{T}_h$. To express q_h^e in terms of u_h we require more notations called edge-wise lifting operators. For each interior edge $e \in \partial\mathcal{T}_h^\circ$, $r^e : [L^2(e)]^d \to \Sigma_h$, $l^e : L^2(e) \to \Sigma_h$, and for each Dirichlet boundary edge $e \in \partial\Omega_D$, $r_D^e : L^2(e) \to \Sigma_h$ are defined as

$$\int_\Omega r^e \tau_h \cdot \sigma_h dx = -\int_e \tau_h \cdot \{\!\{\sigma_h\}\!\}\, d\Gamma, \qquad \tau_h \in [L^2(e)]^d, \forall \sigma_h \in \Sigma_h, \tag{2.2.20a}$$

$$\int_\Omega l^e(w_h) \cdot \sigma_h dx = -\int_e w_h [\![\sigma_h]\!]\, d\Gamma, \qquad w_h \in L^2(e), \forall \sigma_h \in \Sigma_h, \tag{2.2.20b}$$

$$\int_\Omega r_D^e(w_h) \cdot \sigma_h dx = -\int_e w_h \sigma_h \cdot \vec{n} d\Gamma, \qquad w_h \in L^2(e), \forall \sigma_h \in \Sigma_h, \tag{2.2.20c}$$

respectively. Clearly, for all $\tau_h \in \left[L^2\left(\partial\mathcal{T}_h^\circ\right)\right]^d$ and all $w_h \in L^2\left(\partial\mathcal{T}_h^\circ\right)$, we have

$$r\left(\tau_h\right) = \sum_{e\in\partial\mathcal{T}_h^\circ} r^e\left(\tau_h\right), \quad l\left(w_h\right) = \sum_{e\in\partial\mathcal{T}_h^\circ} l^e\left(\tau_h\right), \quad r_D\left(w_h\right) = \sum_{e\in\partial\Omega_D} r_D^e\left(w_h\right).$$

Then, for each edge e of $\partial\mathcal{T}_h$, the expression of q_h^e solely in terms of u_h is

$$q_h^e = \nabla_h u_h + r^e\left(\left[\!\left[u_h\right]\!\right]\right) + l^e\left(C_{12}\cdot\left[\!\left[u_h\right]\!\right]\right) - r_D^e\left(g_D - u_h\right). \tag{2.2.21}$$

The primal form for CDG method is

$$\frac{d}{dt}\int_\Omega u_h v_h dx = aA_h^{\mathrm{CDG}}\left(u_h, v_h\right) = a\left(B_h^{\mathrm{CDG}}\left(u_h, v_h\right) + C^{\mathrm{CDG}}\left(v_h\right)\right), \tag{2.2.22a}$$

in which

$$\begin{aligned} B_h^{\mathrm{CDG}}\left(u_h, v_h\right) = &-\int_\Omega \nabla_h u_h\cdot\nabla_h v_h dx \\ &-\int_\Omega \left(\nabla_h u_h\cdot r\left(\left[\!\left[v_h\right]\!\right]\right) + r\left(\left[\!\left[u_h\right]\!\right]\right)\cdot\nabla_h v_h\right) dx \\ &-\int_\Omega \left(\nabla_h u_h\cdot l\left(C_{12}\cdot\left[\!\left[v_h\right]\!\right]\right) + l\left(C_{12}\cdot\left[\!\left[u_h\right]\!\right]\right)\cdot\nabla_h v_h\right) dx \\ &-\int_\Omega \left(\nabla_h u_h\cdot r_D\left(v_h\right) + r_D\left(u_h\right)\cdot\nabla_h v_h\right) dx \\ &-\sum_{e\in\partial\mathcal{T}_h}\int_\Omega \left(r^e\left(\left[\!\left[u_h\right]\!\right]\right) + l^e\left(C_{12}\cdot\left[\!\left[u_h\right]\!\right]\right) + r_D^e\left(u_h\right)\right) \\ &\qquad\cdot\left(r^e\left(\left[\!\left[v_h\right]\!\right]\right) + l^e\left(C_{12}\cdot\left[\!\left[v_h\right]\!\right]\right) + r_D^e\left(v_h\right)\right) dx \\ &-\int_{\partial\mathcal{T}_h^\circ} C_{11}\left[\!\left[u_h\right]\!\right]\cdot\left[\!\left[v_h\right]\!\right] d\Gamma - \int_{\partial\Omega_D} C_{11}u_h v_h d\Gamma, \end{aligned} \tag{2.2.22b}$$

and

$$\begin{aligned} C_h^{\mathrm{CDG}}\left(v_h\right) = &-\int_{\partial\Omega_D} g_D\vec{n}\cdot\left(\nabla_h v_h + r_D\left(v_h\right)\right) d\Gamma \\ &+\int_{\partial\Omega_D} C_{11}g_D v_h d\Gamma + \int_{\partial\Omega_N} g_N v_h d\Gamma. \end{aligned} \tag{2.2.22c}$$

2.2.4 Order of accuracy and stability

Since the discretization of equation (2.2.1) is as same as the one of the conservation law (2.1.1), the optimal order of accuracy for numerical solution u_h is h^{p+1} in which h is the size of triangulation mesh and p is the order of element-wise approximate polynomials.

The stability of three in four DG methods are also established in this section. The one of CDG method has not been theoretically proven although this method works very well practically. For simplicity we consider only homogeneous boundary conditions, i.e. homogeneous Dirichlet and/or homogeneous Neumann ones. From primal forms of DG methods we have

$$A_h^{\text{Cent}}(u_h, u_h) = -\int_\Omega |\nabla_h u_h + r([\![u_h]\!]) + r_D(u_h)|^2 \, dx, \tag{2.2.23a}$$

$$\begin{aligned} A_h^{\text{IP}}(u_h, u_h) = &-\int_\Omega \left(|\nabla_h u_h|^2 + 2\nabla_h u_h \cdot r([\![u_h]\!]) + 2\nabla_h u_h \cdot r_D(u_h)\right) dx \\ &- \int_{\partial \mathcal{T}_h^\circ} \mu \, |[\![u_h]\!]|^2 \, d\Gamma - \int_{\partial\Omega_D} \mu u_h^2 d\Gamma, \end{aligned} \tag{2.2.23b}$$

$$\begin{aligned} A_h^{\text{LDG}}(u_h, u_h) = &-\int_\Omega |\nabla_h u_h + r([\![u_h]\!]) + l(C_{12} \cdot [\![u_h]\!]) + r_D(u_h)|^2 \, dx \\ &- \int_{\partial \mathcal{T}_h^\circ} C_{11} \, |[\![u_h]\!]|^2 \, d\Gamma - \int_{\partial\Omega_D} C_{11} u_h^2 d\Gamma. \end{aligned} \tag{2.2.23c}$$

For IP method, the interior penalty parameter μ is chosen such that $A_h^{IP} \leqslant 0$ for all u_h [33]. Then, for all three numerical fluxes central, IP, and LDG, we have

$$\frac{d}{dt} \int_\Omega |u_h|^2 \, dx \leqslant 0.$$

This inequality means that the numerical solution using one of the three numerical fluxes are stable. We have the following result.

Proposition 2.2.2. *Three DG methods corresponding to central, IP, and LDG fluxes are stable.*

2.2.5 Numerical results

The semi-discretizing process for (2.2.1) using DG methods leads to the system of algebraic differential equations of the form

$$\mathbf{M}\frac{du_h}{dt} = -a\mathbf{S}u_h. \tag{2.2.24}$$

Here, the matrices $\mathbf{M}$ and $\mathbf{S}$ are respectively called mass and stiffness ones. This system normally has to be integrated implicitly because it requires unacceptably small time step size if the explicit integration methods are involved. Although the implicit solver are able to offer much larger time step size but it is expensive regrading computational cost and memory storage, especially in huge systems of ODEs. Unfortunately, our system (2.2.24) is really huge. Indeed, in d dimensional case, if the domain Ω is partitioned into nE elements and polynomials of degree p are used for approximation on each elements, then the system (2.2.24) consists of $\mathrm{nE} \times \binom{p+d}{d}$ equations. This number is very big if we compute in higher dimensional problems and in refined mesh size of the triangulation. Therefore, we have to make an investigation through all four numerical fluxes to examine several aspects that concern to practical performance.

Sparsity patterns

The first property we would like to look into is sparsity. Stiffness matrix $\mathbf{S}$ is giant with its size of $\left(\mathrm{nE} \times \binom{p+d}{d}\right) \times \left(\mathrm{nE} \times \binom{p+d}{d}\right)$ but it has a lot of zero entries. The number of zero entries in $\mathbf{S}$ depends on which kind of numerical flux in use. Obviously the more zero entries matrix $\mathbf{S}$ has, the cheaper computational cost is.

To discussing the sparsity pattern, we assume that nodal bases are used to span spaces V_h and Σ_h. For illustration purpose, we use the test domain which is made up of four triangles as shown in Figure 2.2.1. On this triangulation, the approximate polynomials is of third degree. The total number of DOFs is 40 corresponding to 10 DOFs per element. The sparsity patterns of central, IP, LDG, and CDG methods are shown in Figure 2.2.2.

From these patterns, the CDG and IP methods are both compact in the sense that they connect only neighboring triangles. Meanwhile the LDG and central methods are non-compact and they give connections between some non-neighboring triangles. In the test domain, the LDG method allows connections between some DOFs in element

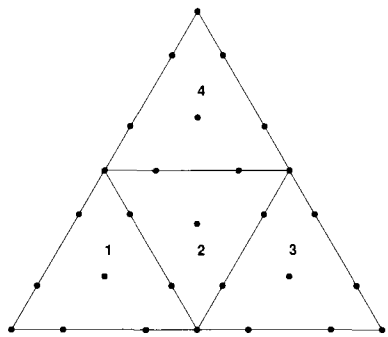


FIGURE 2.2.1: *The test domain for investigating sparsity patterns*

3 and some in element 4. The central method even connects all four elements of the test triangulation. Connections between elements are decided via stabilization terms in primal forms of each methods. For IP and CDG methods, the stabilization terms are respectively

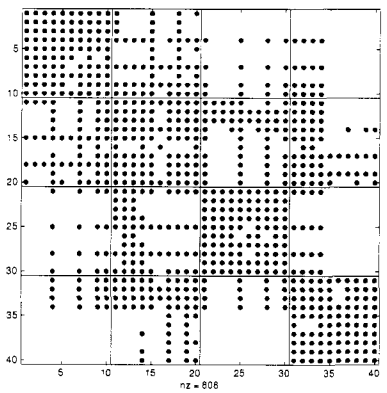
$$\sum_{e\in\partial\mathcal{T}_h^\circ}\int_e \mu\,[\![u_h]\!]\cdot[\![v_h]\!]\,d\Gamma$$

and

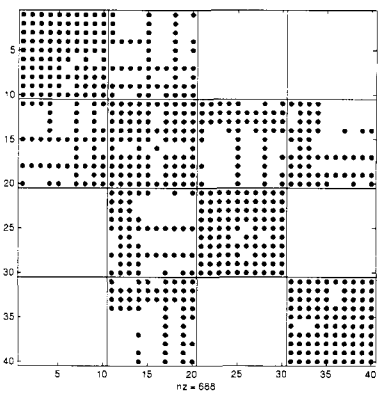
$$\sum_{e\in\partial\mathcal{T}_h^\circ}\int_\Omega \left[r^e\left([\![u_h]\!]\right)+l^e\left(C_{12}\cdot[\![u_h]\!]\right)+r_D^e\left(u_h\right)\right] \cdot\left[r^e\left([\![v_h]\!]\right)+l^e\left(C_{12}\cdot[\![v_h]\!]\right)+r_D^e\left(v_h\right)\right]dx.$$

Summands of these two stabilization terms are non-zero only on the common edge of two adjacent elements and therefore there are only connections between two adjacent elements appearing in stiffness matrix. In contrast, the stabilization terms of central and LDG methods are respectively

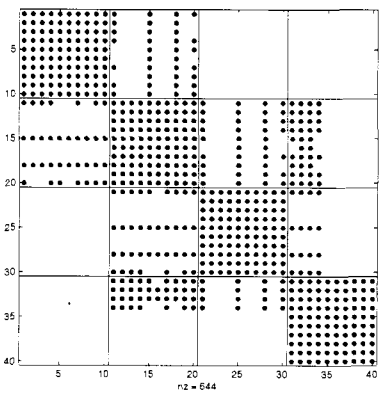
$$\sum_{e_1\in\partial\mathcal{T}_h^\circ}\sum_{e_2\in\partial\mathcal{T}_h^\circ}\int_\Omega \left[r^{e_1}\left([\![u_h]\!]\right)+r_D^{e_1}\left(u_h\right)\right]\cdot\left[r^{e_2}\left([\![v_h]\!]\right)+r_D^{e_2}\left(v_h\right)\right]dx$$



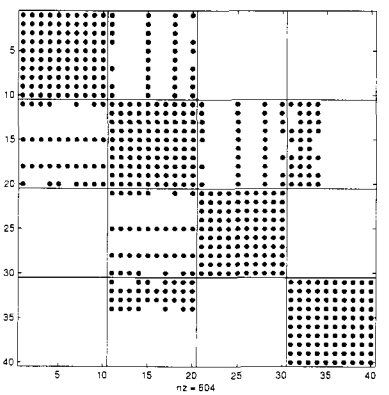
(A) *Central method with 808 non-zeros entries.*



(B) *IP method with 688 non-zeros entries.*



(C) *LDG method with 644 non-zeros entries.*



(D) *CDG method with 604 non-zeros entries.*

FIGURE 2.2.2: *Sparsity patterns of four kinds of numerical fluxes.*

and

$$\sum_{e_1 \in \partial \mathcal{T}_h^\circ} \sum_{e_2 \in \partial \mathcal{T}_h^\circ} \int_\Omega \left[r^{e_1}\left(\llbracket u_h \rrbracket\right) + l^{e_1}\left(C_{12} \cdot \llbracket u_h \rrbracket\right) + r_D^{e_1}\left(u_h\right) \right] \cdot \left[r^{e_2}\left(\llbracket v_h \rrbracket\right) + l^{e_2}\left(C_{12} \cdot \llbracket v_h \rrbracket\right) + r_D^{e_2}\left(v_h\right) \right] dx.$$

It can be clearly seen in these terms that there are connections between arbitrary two elements and this fact results the non-compactness of stiffness matrices of central and LDG methods.

Moreover, the stabilization term of CDG allows only connections between facial nodes on each face of element and all nodes of adjacent element sharing this face. The IP methods connects facial nodes of each element to all nodes of its adjacent elements. Therefore the non-zeros entries of $\mathbf{S}$ concerning to CDG method are less than those of IP method.

Matrix storage

In order to estimate memory requirement for storing stiffness matrices of four methods, we consider simplex elements in d dimensions having $d+1$ adjacent elements. If approximate polynomials of degree p are in use, then the number of DOFs on each element is $\mathtt{nP} = \binom{p+d}{d}$ and the number of DOFs on each face of element is $\mathtt{nfP} = \binom{p+d-1}{d-1}$.

For the CDG method, it requires to store $\mathtt{nP}^2$ entries for a diagonal block and $d+1$ off-diagonal blocks. Each of off-diagonal blocks has $\mathtt{nP} \times \mathtt{nfP}$ non-zero entries. In total the memory to store connections of a single interior element is

$$\mathtt{mem}_{\mathtt{CDG}} = \mathtt{nP}^2 + (d+1) \times \mathtt{nP} \times \mathtt{nfP}. \tag{2.2.25}$$

The IP method also has $d+1$ off-diagonal blocks but each of these blocks has $\mathtt{nfP} \times \mathtt{nP} + \mathtt{nfP} \times (\mathtt{nP} - \mathtt{nfP})$. It then results

$$\mathtt{mem}_{\mathtt{IP}} = \mathtt{nP}^2 + (d+1) \times \mathtt{nfP} \times (2 \times \mathtt{nP} - \mathtt{nfP}). \tag{2.2.26}$$

The pattern of LDG is similar to that of CDG plus some additional non-local connections. The number of non-adjacent elements connections depends on the mesh and switch. This number denoted as a is equal to zero in one dimensional domain. For higher dimensional domain, in average, $a \approx 1$ for $d = 2$ and $a = 2$ for $d = 3$. The total number

of non-zeros is then

$$\mathtt{mem}_{\mathtt{LDG}} = \mathtt{nP}^2 + (d+1) \times \mathtt{nfP} \times \mathtt{nP} + a\mathtt{nfP}^2. \tag{2.2.27}$$

For the central method, its number of non-zeros entries is relatively big in comparison with other methods. The connections of central method are between not only adjacent elements but also between elements having common vertices in two dimensional case and having common vertices or edges in three dimensional case. This number of connections denoted by β depends on mesh. If uniform simplexes are used in triangulating domain, then $\beta = 0$ for $d = 1$, $\beta = 12$ for $d = 2$. The total number of non-zeros in this case is

$$\mathtt{mem}_{\mathtt{Cent}} = \mathtt{nP}^2 + (d+1) \times \mathtt{nfP} \times (2 \times \mathtt{nP} - \mathtt{nfP}) + \beta \mathtt{nfP}^2. \tag{2.2.28}$$

For three dimensional case, $d = 3$, this number has not been determined yet.

The memory requirements for $d = 1, 2$ and $p = 1, \ldots, 5$ are shown in Table 2.2.1. We can see that the central method consume much more memory than other methods. The CDG method has the lowest memory requirements. In two dimensional case with polynomials of degree $p = 3$, memory usage of the LDG and IP are more 7% and 32% respectively in compare to CDG method.

Dimension	Method	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
$d=1$	Central	10	19	30	43	58
	IP	10	19	30	43	58
	LDG	8	15	24	35	48
	CDG	8	15	24	35	48
$d=2$	Central	81	225	484	900	1521
	IP	33	117	292	600	1089
	LDG	31	99	236	475	855
	CDG	27	90	220	450	819

TABLE 2.2.1: *Memory requirements per simplex interior element for central, IP, LDG, and CDG methods.*

SPECTRAL RADIUS

Here, we compare spectral radius of stiffness matrix generated by four DG methods. The spectral radius is very important to determine stiffness of the semi-discretized diffusion equations and goodness of the approximation of discretized Laplacian operator to the original one.

We consider the following test problem

$$\frac{\partial u}{\partial t} = 0.1\Delta u, \qquad (x, y) \in \Omega \subset \mathbb{R}^2, \tag{2.2.29a}$$

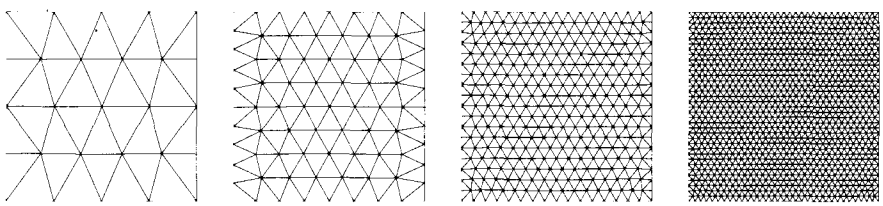
$$\frac{\partial u}{\partial \vec{n}}|_{\partial\Omega} = 0, \tag{2.2.29b}$$

$$u|_{t=0} = \cos(2\pi x)\cos(2\pi y). \tag{2.2.29c}$$

The computational domian Ω is a unit reactangle $(0, 1)^2$ and the time interval is $(0, 0.1)$. The exact solution is

$$u(x, y, t) = \exp(-0.8\pi^2 t)\, u(x, y, 0). \tag{2.2.29d}$$

The domain Ω is triangulated by unstructured mesh with triangles of variable sizes which are $h \in \{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}\}$. The four triangulations are shown in Figure 2.2.3. On each triangulation's mesh size, we use polynomials of degree from 1 to 4 to approximate the exact solution of (2.2.29).



(A) $h = 2^{-2}$. (B) $h = 2^{-3}$. (C) $h = 2^{-4}$. (D) $h = 2^{-5}$.

FIGURE 2.2.3: *Triangulation meshes of the domain* $\Omega = [0, 1]^2$.

We have total 16 test problems of the form (2.2.24). Spectral radius of DG methods are determined as the maximum of absolute value of eigenvalues of stiffness matrix **S**. Sixty four spectral radius corresponding to four DG methods and 16 test problems are

Degree	Method	$h = 2^{-2}$	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$
$p = 1$	Central	1.1201e+02	4.4588e+02	1.7908e+03	7.1548e+03
	IP	4.0804e+02	1.8005e+03	6.4615e+03	2.5868e+04
	LDG	2.4632e+02	9.7945e+02	4.0072e+03	1.6029e+04
	CDG	2.4255e+02	9.5870e+02	3.9262e+03	1.5723e+04
$p = 2$	Central	7.6687e+02	3.1213e+03	1.2326e+04	4.9210e+04
	IP	2.7301e+03	1.2707e+04	4.1947e+04	1.6817e+05
	LDG	1.6621e+03	6.2242e+03	2.6491e+04	1.0579e+05
	CDG	1.6246e+03	6.2301e+03	2.5686e+04	1.0257e+05
$p = 3$	Central	1.9955e+03	8.0144e+03	3.1570e+04	1.2634e+05
	IP	7.4254e+03	3.4812e+04	1.1388e+05	4.5670e+05
	LDG	4.6647e+03	1.7703e+04	7.3808e+04	2.9470e+05
	CDG	4.6172e+03	1.7704e+04	7.2837e+04	2.9120e+05
$p = 4$	Central	5.0408e+03	1.9745e+04	7.9593e+04	3.1879e+05
	IP	1.6008e+04	7.4910e+04	2.4561e+05	9.8498e+05
	LDG	1.0438e+04	3.9921e+04	1.6468e+05	6.5750e+05
	CDG	1.0390e+04	3.9921e+04	1.6373e+05	6.5647e+05

TABLE 2.2.2: *Spectral radius of DG methods, scaled by* $(h/p)^2$.

shown in Table 2.2.2.

Observing spectral radius of DG methods, we realize that the semi-discretized ODEs of the linear diffusion (2.2.29) is very stiff. Explicit methods are unable to apply efficiently to solve these ODEs because if one does those, time step size is required to be unacceptably small to avoid spurious oscillations. It is preferred to employ A- or L- stable ODE solvers for these problems.

Table 2.2.2 shows that the spectral radius of LDG and CDG methods are almost the same while IP method possesses double size of spectral radius and central method gives 50% smaller spectral radius. It means that the IP method approximates physically the Laplacian operator better than other ones. Eigenvalues of discretized Laplacian operators of central, IP, and LDG methods are all negative that is high agreement with the stability of these methods proved in Section 2.2.4. Although the stability of CDG methods has not been proven theoretically yet, but CDG method also gives all negative eigenvalues. This fact suggests that CDG method is applicable to parabolic equations.

Order of accuracy

To test the spatial order of accuracy, we semi-discretize (2.2.29) using all four DG methods which are central, IP, LDG, and CDG. For integration in time, we use a L-stable singly diagonal implicit Runge-Kutta method which is of five-stage, fourth-order and denote such method as SDIRK54. The Butcher table of SDIRK54 method is given in Table 2.2.3. Time step size is taken to be small such that temporal error has not any remarkable effect to the error estimation. Absolute errors and estimated order of accuracy of four DG methods are given in Tables 2.2.4 and 2.2.5, respectively.

$1/4$	$1/4$	0	0	0	0
$3/4$	$1/2$	$1/4$	0	0	0
$11/20$	$17/50$	$-1/25$	$1/4$	0	0
$1/2$	$371/1360$	$-137/2720$	$15/544$	$1/4$	0
1	$25/24$	$-49/48$	$125/16$	$-85/12$	$1/4$
	$25/24$	$-49/48$	$125/16$	$-85/12$	$1/4$
	$59/48$	$-17/96$	$225/32$	$-85/12$	0

Table 2.2.3: *The Butcher table of L-stable SDIRK54.*

By order of accuracy, all four DG methods generally attain the optimal order which is $p+1$ if polynomials of degree p are in use. The central method approximates quite well in coarse triangulations but it becomes worse than other ones in more refined meshes. The LDG and CDG methods' behaviors are quite similar but CDG is slightly better than LDG. In the tough test with $h = 2^{-5}$ and $p = 4$, IP method is the only one maintaining the optimal order of accuracy. Another ones, especially LDG method, lose their optimal orders. It hints that the constant of error estimation of LDG method is smaller than those of other methods so that the temporal error affects significantly the spatial error.

Computational speed

To compare computational speeds, we remove the time step size limitation and let SDIRK54 solver decide automatically time step size.

Central method is the slowest because of its too big memory storage. IP method which requires moderately bigger memory storage than CDG but its computational times

Degree	Method	$h = 2^{-2}$	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$
$p = 1$	Central	3.6252e-01	1.2317e-01	2.7110e-02	6.6902e-03
	IP	4.2714e-01	1.5577e-01	3.4344e-02	8.4999e-03
	LDG	3.5145e-01	1.2126e-01	2.6074e-02	6.3947e-03
	CDG	3.5009e-01	1.2074e-01	2.6017e-02	6.3840e-03
$p = 2$	Central	3.8081e-02	4.1312e-03	3.7965e-04	4.4940e-05
	IP	4.4674e-02	4.9356e-03	4.7726e-04	5.6270e-05
	LDG	3.9988e-02	4.3717e-03	4.3638e-04	5.2448e-05
	CDG	3.9299e-02	4.2693e-03	4.2205e-04	5.0548e-05
$p = 3$	Central	3.2573e-03	3.5454e-04	2.0845e-05	1.9913e-06
	IP	3.8170e-03	3.8978e-04	1.7182e-05	9.8640e-07
	LDG	3.4985e-03	3.4729e-04	1.5134e-05	8.8061e-07
	CDG	3.4243e-03	3.4237e-04	1.4999e-05	8.6668e-07
$p = 4$	Central	5.9288e-04	1.7031e-05	3.5398e-07	1.6640e-08
	IP	6.6159e-04	1.8689e-05	4.0353e-07	1.2651e-08
	LDG	6.0212e-04	1.6576e-05	3.6414e-07	6.1387e-08
	CDG	5.9411e-04	1.6351e-05	3.6021e-07	2.1292e-08

TABLE 2.2.4: *Absolute errors of numerical solutions at* $t = 0.1$.

Degree	Method	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$
$p = 1$	Central	1.5574	2.1838	2.0187
	IP	1.4553	2.1813	2.0146
	LDG	1.5352	2.2174	2.0277
	CDG	1.5358	2.2144	2.0269
$p = 2$	Central	3.2045	3.4438	3.0786
	IP	3.1781	3.3704	3.0843
	LDG	3.1933	3.3245	3.0566
	CDG	3.2024	3.3385	3.0617
$p = 3$	Central	3.1996	4.0882	3.3879
	IP	3.2917	4.5037	4.1225
	LDG	3.3325	4.5203	4.1032
	CDG	3.3222	4.5126	4.1132
$p = 4$	Central	5.1215	5.5884	4.4109
	IP	5.1457	5.5333	4.9954
	LDG	5.1829	5.5084	2.5685
	CDG	5.1833	5.5044	4.0805

TABLE 2.2.5: *Estimated order of accuracy of numerical solutions at* $t = 0.1$.

Degree	Method	$h = 2^{-2}$	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$
$p = 3$	Central	2.9689e-01	5.2601e-01	1.9552e+00	9.6656e+00
	IP	3.2532e-01	4.8565e-01	1.5597e+00	7.9146e+00
	LDG	3.2262e-01	4.9977e-01	1.3777e+00	6.0831e+00
	CDG	3.3647e-01	5.2028e-01	1.3872e+00	7.1063e+00
$p = 4$	Central	3.3572e-01	6.7468e-01	3.6038e+00	2.0293e+01
	IP	3.7148e-01	6.4973e-01	3.1498e+00	1.5580e+01
	LDG	3.3006e-01	6.0632e-01	2.4728e+00	1.3352e+01
	CDG	3.4358e-01	6.1517e-01	2.4095e+00	1.3918e+01

TABLE 2.2.6: *Computational time of DG methods solving problem* (2.2.29)

are not different so much from CDG method. LDG method is still faster than CDG in most test problems. One of the reasons for unimpressive CDG's computational time is the cost to compute the CDG fluxes is more expensive than others.

The poor performance of central method is caused not only by memory storage but also by its stiffness ratio. The stiffness ratio of the semi-discretize problem (2.2.24) is defined by

$$\text{Stiffness ratio} = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

in which $\lambda_{\max}$ is the largest magnitude eigenvalue of $\mathbf{S}$ and $\lambda_{\min}$ is the smallest magnitude eigenvalue of $\mathbf{S}$. Table 2.2.7 gives stiffness ratios of central and IP methods in test problems.

Degree	Method	$h = 2^{-2}$	$h = 2^{-3}$	$h = 2^{-4}$	$h = 2^{-5}$
$p = 1$	Central	5.7461e+18	7.3768e+18	2.5321e+19	9.3817e+19
	IP	1.1704e+18	6.1610e+18	6.9141e+19	1.2333e+20
$p = 2$	Central	3.5176e+18	1.4105e+20	2.3545e+20	1.0283e+21
	IP	1.5680e+18	1.7780e+19	6.5995e+19	2.9790e+20
$p = 3$	Central	3.6429e+18	2.1920e+19	1.0082e+20	2.6275e+21
	IP	7.2017e+18	5.1068e+19	4.3875e+19	1.8908e+20
$p = 4$	Central	7.6642e+18	1.4556e+19	6.2022e+19	3.9855e+20
	IP	1.3079e+19	2.9976e+19	8.4438e+19	4.5561e+20

TABLE 2.2.7: *Stiffness ratios of central and LDG methods.*

We can see that even the spectral radius of central method is about a quarter of LDG method but its stiffness ratio is roughly equal to that of LDG method. It means that the

semi-discretized equation using central fluxes contains very slow transients in compare to that equation of using LDG fluxes. These transients decay slowly and therefore the time step size must be smaller even when $t \gg 0$ in order to maintain accuracy.

To see this phenomena more clear, we extend the time interval to $(0, 0.5)$ and plot out in Figure 2.2.4 the time step size of four DG methods during solving processes. Because of smaller time step size, central method needs 42 steps to complete the integration process while CDG and IP methods finish after 27 steps and this number is 26 for LDG method.

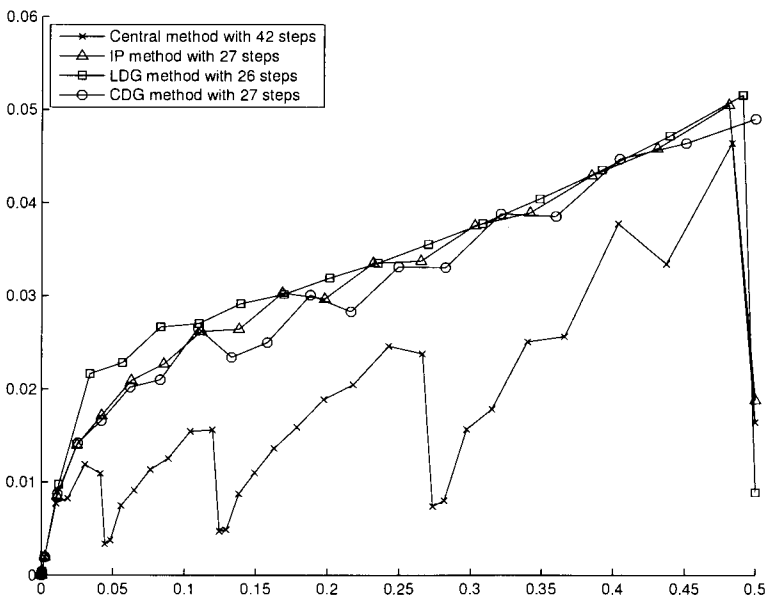


FIGURE 2.2.4: *Time step size of four DG methods.*

CONCLUSION

After investigating four DG methods, we can conclude that

- Despite of simplicity, the central method is the worst in every aspect. It is small spectral radius but large stiffness ratio, slow, and ravenous in memory storage.
- The CDG has as many pretty properties as LDG method but its stability must be proven theoretically.

- The IP method is not so good as LDG but its compactness is a big advantage if one wants to employ it to construct a parallel solver of parabolic equation.
- The LDG is the best choice so far for conventional computing.

2.3 Advection-Diffusion-Reaction equations

In this section, we present how to apply DG methods to the general advection-diffusion-reaction (ADR) problem. We consider a well-posed ADR equation of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot F^{\mathrm{inv}}(u) = \nabla \cdot F^{\mathrm{vis}}(u, \nabla u) + F^{\mathrm{src}}(u) \tag{2.3.1}$$

in a domain $\Omega \subset \mathbb{R}^d$ with state variables u, inviscid flux function F^{inv}, viscous flux function F^{vis}, and source term F^{src}. The boundary condition and initial value are given.

First, we transfer this equation which is of second order spatial derivatives of u to a system of first order by introducing additional variables $q = \nabla u$:

$$\frac{\partial u}{\partial t} = -\nabla \cdot F^{\mathrm{inv}}(u) + \nabla \cdot F^{\mathrm{vis}}(u, q) + F^{\mathrm{src}}(u), \tag{2.3.2a}$$

$$q = \nabla u. \tag{2.3.2b}$$

Next, we consider a triangulation of the spatial domain Ω denoted by $\mathcal{T}_h$ which is of size h. On this triangulation we define approximation finite element broken sapces V_h and Σ_h as same as in (2.2.3) and (2.2.4). The DG formulations for (2.3.2) read: Find $u_h \in V_h$ and $q_h \in \Sigma_h$ such that for all elements $D^k \in \mathcal{T}_h$, for all $v_h \in V_h$, and for all $\sigma_h \in \Sigma_h$, we have

$$\begin{aligned}\frac{d}{dt}\int_{D^k} u_h v_h dx = &\int_{D^k} F^{\mathrm{inv}}(u_h) \cdot \nabla v_h dx - \int_{\partial D^k} \widehat{F^{\mathrm{inv}}}(u_h) \cdot \vec{n} v_h d\Gamma \\ &- \int_{D^k} F^{\mathrm{vis}}(u_h, q_h) \cdot \nabla v_h dx + \int_{\partial D^k} \widehat{F^{\mathrm{vis}}}(u_h, q_h) \cdot \vec{n} v_h d\Gamma \\ &+ \int_{D^k} F^{\mathrm{src}}(u_h) v_h dx, \end{aligned} \tag{2.3.3a}$$

$$\int_{D^k} q_h \cdot \sigma_h dx = -\int_{D^k} u_h \nabla \cdot \sigma_h dx + \int_{\partial D^k} \widehat{u}_h \vec{n} \cdot \sigma_h d\Gamma. \tag{2.3.3b}$$

Here, the numerical fluxes $\widehat{F^{\mathrm{inv}}}$, $\widehat{F^{\mathrm{vis}}}$, and $\widehat{u}_h$ are approximations to F^{inv}, F^{vis}, and u_h,

respectively, on the boundary ∂D^k. As investigated so far, the numerical inviscid fluxes $\widehat{F^{\mathrm{inv}}}$ are approximated using consistency ones of which some of them are proposed in (2.1.4). For the numerical viscous fluxes $\widehat{F^{\mathrm{vis}}}$, any of four numerical fluxes in Section 2 is a good candidate.

To obtain (2.3.3) as an original ODEs system, in fact, we compute (2.3.3) directly rather relying on primal forms. We suppose that $\left\{v_j^k\right\}_{j=1,\ldots,\mathrm{nP}}$ is a d-dimensional polynomial basis of degree p of V_h on each element D^k. Then the approximate solution u_h can be expressed as

$$u_h(x,t) = \bigoplus_{k=1}^{\mathrm{nE}} u_h(x,t)\big|_{D^k} \quad \text{in which} \quad u_h(x,t)\big|_{D^k} = \sum_{j=1}^{\mathrm{nP}} u_i^k(t)\, v_i^k(x). \quad (2.3.4a)$$

Basis on each element D^k of Σ_h derived from one of V_h is $\left\{\sigma_j^{x_i,k}\right\}_{j=1,\ldots,\mathrm{nP}}^{i=1,\ldots,d}$ in which

$$\sigma_j^{x_i,k} = \left[0,\ldots,0,v_j^k,0,\ldots,0\right]^T, \quad v_j^k \text{ is at the } j(\mathrm{th}) \text{ position.}$$

If $q_h = \left[q_h^{x_1},\ldots,q_h^{x_d}\right]$, then for each i from 1 to d, $q_h^{x_i}$ is presented similarly as u_h:

$$q_h^{x_i}(x,t) = \bigoplus_{k=1}^{\mathrm{nE}} q_h^{x_i}(x,t)\big|_{D^k} \quad \text{in which} \quad q_h^{x_i}(x,t)\big|_{D^k} = \sum_{j=1}^{\mathrm{nP}} q_j^{x_i,k}(t)\, v_j^k(x). \quad (2.3.4b)$$

Therefore u_h and $q_h^{x_i}$, $i = 1,\ldots,d$ on each element D^k are represented by coefficients $\left[u_j^k\right]_{j=1,\ldots,\mathrm{nP}}$ and $\left[q_j^{x_i,k}\right]_{j=1,\ldots,\mathrm{nP}}$. For this reason we coincide $u_h\big|_{D^k}$ with column vector $\left[u_j^k\right]_{j=1,\ldots,\mathrm{nP}}$ and u_h with vector $\left[u_1^1,\ldots,u_{\mathrm{nP}}^1,\ldots,u_1^{\mathrm{nE}},\ldots,u_{\mathrm{nP}}^{\mathrm{nE}}\right]^T$. The same notations are applied to $q_h^{x_i}\big|_{D^k}$ and $q_h^{x_i}$.

For each $i = 1,\ldots,d$, the computation of $q_h^{x_i}$ reads: for $l = 1,\ldots,\mathrm{nP}$,

$$\int_{D^k} \sum_{j=1}^{\mathrm{nP}} q_j^{x_i,k} v_j^k v_l^k dx = -\int_{D^k} \sum_{j=1}^{\mathrm{nP}} u_j^k v_j^k \frac{\partial v_l^k}{\partial x_i} dx + \int_{\partial D^k} \sum_{j=1}^{\mathrm{nP}} \widehat{u}_j^k v_j^k \big|_{\partial D^k} \vec{n}^{x_i} v_l^k d\Gamma. \quad (2.3.5)$$

Here, the numerical fluxes $\widehat{u}_h$ at ∂D^k is approximated by $\sum_{j=1}^{\mathrm{nP}} \widehat{u}_j^k v_j^k\big|_{\partial D^k}$ and n^{x_i} is the i(th) element of the outward normal $\vec{n}$ to element D^k. If we denote local mass matrix

M^k, local stiffness matrix $S^{x_i,k}$, and lifting matrix $L^{x_i,k}$ whose entries are

$$M^k_{l,j} = \int_{D^k} v^k_l v^k_j dx, \quad S^{x_i,k}_{l,j} = \int_{D^k} \frac{\partial v^k_l}{\partial x_i} v^k_j dx, \quad L^{x_i,k}_{l,j} = \int_{\partial D^k} \bar{n}^{x_i} v^k_l v^k_j d\Gamma,$$

then the matrix form of (2.3.5) is

$$M^k q^{x_i,k} = -S^{x_i,k} u^k + L^{x_i,k} \widehat{u}^k, \qquad i = 1, \ldots, d. \tag{2.3.6}$$

Here, $\widehat{u}^k$ is coincided with column vector $\left[\widehat{u}^k_1, \ldots, \widehat{u}^k_{\mathrm{nP}}\right]^T$. It is worth to note that M^k is symmetric and invertible.

Applying the same manner as (2.3.3a), we have

$$\begin{aligned} M^k \frac{du^k_h}{dt} = & \sum_{i=1}^{d} S^{x_i,k} \left(F^{\mathrm{inv}}\right)^{x_i,k} - \sum_{i=1}^{d} L^{x_i,k} \left(\widehat{F^{\mathrm{inv}}}\right)^{x_i,k} \\ & - \sum_{i=1}^{d} S^{x_i,k} \left(F^{\mathrm{vis}}\right)^{x_i,k} + \sum_{i=1}^{d} L^{x_i,k} \left(\widehat{F^{\mathrm{vis}}}\right)^{x_i,k} \\ & + M^k \left(F^{\mathrm{src}}\right)^k, \end{aligned} \tag{2.3.7}$$

in which

$$\left(F^{\mathrm{inv}}\right)^{x_i}\Big|_{D^k} \approx \sum_{j=1}^{\mathrm{nP}} \left(F^{\mathrm{inv}}\right)^{x_i,k}_j v^k_j, \qquad \left(F^{\mathrm{inv}}\right)^{x_i,k} = \left[\left(F^{\mathrm{inv}}\right)^{x_i,k}_1, \ldots, \left(F^{\mathrm{inv}}\right)^{x_i,k}_{\mathrm{nP}}\right]^T,$$

$$\left(\widehat{F^{\mathrm{inv}}}\right)^{x_i}\Big|_{D^k} \approx \sum_{j=1}^{\mathrm{nP}} \left(\widehat{F^{\mathrm{inv}}}\right)^{x_i,k}_j v^k_j, \quad \left(\widehat{F^{\mathrm{inv}}}\right)^{x_i,k} = \left[\left(\widehat{F^{\mathrm{inv}}}\right)^{x_i,k}_1, \ldots, \left(\widehat{F^{\mathrm{inv}}}\right)^{x_i,k}_{\mathrm{nP}}\right]^T,$$

$$\left(F^{\mathrm{vis}}\right)^{x_i}\Big|_{D^k} \approx \sum_{j=1}^{\mathrm{nP}} \left(F^{\mathrm{vis}}\right)^{x_i,k}_j v^k_j, \qquad \left(F^{\mathrm{inv}}\right)^{x_i,k} = \left[\left(F^{\mathrm{vis}}\right)^{x_i,k}_1, \ldots, \left(F^{\mathrm{vis}}\right)^{x_i,k}_{\mathrm{nP}}\right]^T,$$

$$\left(\widehat{F^{\mathrm{vis}}}\right)^{x_i}\Big|_{D^k} \approx \sum_{j=1}^{\mathrm{nP}} \left(\widehat{F^{\mathrm{vis}}}\right)^{x_i,k}_j v^k_j, \quad \left(\widehat{F^{\mathrm{vis}}}\right)^{x_i,k} = \left[\left(\widehat{F^{\mathrm{vis}}}\right)^{x_i,k}_1, \ldots, \left(\widehat{F^{\mathrm{vis}}}\right)^{x_i,k}_{\mathrm{nP}}\right]^T,$$

and

$$F^{\mathrm{src}}\Big|_{D^k} \approx \sum_{j=1}^{\mathrm{nP}} \left(F^{\mathrm{src}}\right)^k_j v^k_j, \qquad \left(F^{\mathrm{src}}\right)^k = \left[\left(F^{\mathrm{src}}\right)^k_1, \ldots, \left(F^{\mathrm{src}}\right)^k_{\mathrm{nP}}\right]^T.$$

Gathering all equations (2.3.7) on all elements D^k, we arrive the following ODEs

$$\begin{aligned} M\frac{du_h}{dt} = &\sum_{i=1}^{d} S^{x_i}\left(F^{\mathtt{inv}}\right)^{x_i} - \sum_{i=1}^{d} L^{x_i}\left(\widehat{F^{\mathtt{inv}}}\right)^{x_i} \\ &- \sum_{i=1}^{d} S^{x_i}\left(F^{\mathtt{vis}}\right)^{x_i} + \sum_{i=1}^{d} L^{x_i}\left(\widehat{F^{\mathtt{vis}}}\right)^{x_i} + MF^{\mathtt{src}}. \end{aligned} \tag{2.3.8}$$

Here, M, S^{x_i}, L^{x_i} are block-diagonal matrices in which their on-diagonal blocks are M^k, $S^{x_i,k}$, and $L^{x_i,k}$, respectively. Vectors $\left(F^{\mathtt{inv}}\right)^{x_i}$, $\left(\widehat{F^{\mathtt{inv}}}\right)^{x_i}$, $\left(F^{\mathtt{vis}}\right)^{x_i}$, $\left(\widehat{F^{\mathtt{vis}}}\right)^{x_i}$, and $F^{\mathtt{src}}$ are defined as

$$\begin{aligned} \left(F^{\mathtt{inv}}\right)^{x_i} &= \left[\left(\left(F^{\mathtt{inv}}\right)^{x_i,1}\right)^T, \ldots, \left(\left(F^{\mathtt{inv}}\right)^{x_i,\mathtt{nE}}\right)^T\right]^T, \\ \left(\widehat{F^{\mathtt{inv}}}\right)^{x_i} &= \left[\left(\left(\widehat{F^{\mathtt{inv}}}\right)^{x_i,1}\right)^T, \ldots, \left(\left(\widehat{F^{\mathtt{inv}}}\right)^{x_i,\mathtt{nE}}\right)^T\right]^T, \\ \left(F^{\mathtt{vis}}\right)^{x_i} &= \left[\left(\left(F^{\mathtt{vis}}\right)^{x_i,1}\right)^T, \ldots, \left(\left(F^{\mathtt{vis}}\right)^{x_i,\mathtt{nE}}\right)^T\right]^T, \\ \left(\widehat{F^{\mathtt{vis}}}\right)^{x_i} &= \left[\left(\left(\widehat{F^{\mathtt{vis}}}\right)^{x_i,1}\right)^T, \ldots, \left(\left(\widehat{F^{\mathtt{vis}}}\right)^{x_i,\mathtt{nE}}\right)^T\right]^T, \\ F^{\mathtt{src}} &= \left[\left(\left(F^{\mathtt{src}}\right)^1\right)^T, \ldots, \left(\left(F^{\mathtt{src}}\right)^{\mathtt{nE}}\right)^T\right]^T. \end{aligned}$$

The semi-discretized system of ODEs (2.3.8) is final form we obtain after the first phase of our full discretization process. The second phase of the process will be discussed in the next chapter.

3

Rosenbrock strong-stability preserving methods

The second phase of discretization process for ADR equations is considered in this chapter. In the last chapter, spatial discretization using DG methods for the ADR equation results a system of ODEs. There are many barriers for solving this system. Three parts in the system derived respectively from advection, diffusion, and reaction terms of the original ADR equations possess different properties which require very different temporal discretization approach to deal with. The spatial order of accuracy would be high if approximate polynomials of high order is used. That fact demands temporal order of accuracy must be adequate for. Moreover, the size of this system is very big so that a fast solver is necessary. All of these matters are able to be overcome with a new class of methods, which we will call the Rosenbrock strong stability preserving (Ross-SSP) method, introduced in this chapter.

3.1 *N*-TREE THEORY

Methods introduced in later sections need some conditions for their coefficients to achieve the specific order of accuracy. Order conditions of numerical methods for conventional

ODEs is quite confused and it becomes amazingly complex if the ODE are partitioned in several terms and each of these terms is treated with distinguished manner. Therefore we present in this section N-theory which is a powerful tool to possibly list order conditions of a solver for our semi-discretized system.

We consider a decomposed system of the form

$$\frac{dy}{dt} = f(y) = \sum_{\nu\in\Lambda} f^{[\nu]}(y)\,, \tag{3.1.1}$$

in which Λ is a finite index-set and $f = \sum_{\nu\in\Lambda} f^{[\nu]} : U \to \mathbb{R}^d$ are supposed to be smooth and defined in a $\mathbb{R}^d$. To determine order conditions, one can use the Taylor expansion of the solutions of (3.1.1) in term of $f^{[\nu]}$ which is

$$\frac{d^2y^i}{dt^2} = \sum_{\nu,\mu\in\Lambda}\sum_{j=1}^{n} f_j^{[\nu],i} f^{[\mu],j}, \tag{3.1.2a}$$

$$\frac{d^3y^i}{dt^3} = \sum_{\nu,\mu,\lambda\in\Lambda}\sum_{j,k=1}^{n} \left(f_{j,k}^{[\nu],i} f^{[\mu],j} f^{[\lambda],k} + f_j^{[\nu],i} f_k^{[\mu],j} f^{[\lambda],k}\right), \tag{3.1.2b}$$

in which

$$f_j^{[\nu],i} := \frac{\partial f^{[\nu],i}}{\partial x_j}, \quad f_{j,k}^{[\nu],i} := \frac{\partial^2 f^{[\nu],i}}{\partial x_j \partial x_k}.$$

In a case of $\Lambda = \{\nu, \mu\}$, we have

$$\begin{aligned}
\frac{dy^i}{dt} &= f^{[\nu],i}(y) + f^{[\mu],i}(y)\,,\\
\frac{d^2y^i}{dt^2} &= \sum_{j=1}^{n}\left(f_j^{[\nu],i} f^{[\nu],j} + f_j^{[\nu],i} f^{[\mu],j} + f_j^{[\mu],i} f^{[\nu],j} + f_j^{[\mu],i} f^{[\mu],j}\right),\\
\frac{d^3y^i}{dt^3} &= \sum_{j,k=1}^{n}\Big(f_{j,k}^{[\nu],i} f^{[\nu],j} f^{[\nu],k} + f_{j,k}^{[\nu],i} f^{[\nu],j} f^{[\mu],k} + f_j^{[\nu],i} f_k^{[\nu],j} f^{[\nu],k}\\
&\quad + f_j^{[\nu],i} f_k^{[\nu],j} f^{[\mu],k}\\
&\quad + f_{j,k}^{[\nu],i} f^{[\mu],j} f^{[\nu],k} + f_{j,k}^{[\nu],i} f^{[\mu],j} f^{[\mu],k} + f_j^{[\nu],i} f_k^{[\mu],j} f^{[\nu],k}\\
&\quad + f_j^{[\nu],i} f_k^{[\mu],j} f^{[\mu],k} + \dots\Big).
\end{aligned}$$

We associate to the summands on the right hand a graph whose nodes are double

labelled. Such graphs are called N-trees.

2-tree	Elementary differential
$\emptyset$	y^i
ν, i	$f^{[\nu],i}$
μ, i	$f^{[\mu],i}$
$\nu, j \quad \mu, j$ $\vert \quad \vert$ $\nu, i \quad , \quad \nu, i$	$f_j^{[\nu],i} f^{[\nu],j}$, $f_j^{[\nu],i} f^{[\mu],j}$
$\nu, j \quad \mu, j$ $\vert \quad \vert$ $\mu, i \quad , \quad \mu, i$	$f_j^{[\mu],i} f^{[\nu],j}$, $f_j^{[\mu],i} f^{[\mu],j}$
$\nu, j \quad \nu, k \quad \mu, j \quad \nu, k$ $\diagdown \diagup \quad \diagdown \diagup$ $\nu, i \quad , \quad \nu, i$	$f_{j,k}^{[\nu],i} f^{[\nu],j} f^{[\nu],k}$, $f_{j,k}^{[\nu],i} f^{[\mu],j} f^{[\nu],k}$
$\nu, k \quad \mu, k$ $\vert \quad \vert$ $\nu, j \quad \nu, j$ $\vert \quad \vert$ $\nu, i \quad , \quad \nu, i$	$f_j^{[\nu],i} f_k^{[\nu],j} f^{[\nu],k}$, $f_j^{[\nu],i} f_k^{[\nu],j} f^{[\mu],k}$
...	...

TABLE 3.1.1: *Double labelled graphs and corresponding elementary differentials.*

3.1.1 N-TREES

Definition 3.1.1. *Let q be a positive integer and A_q be an order chain of q indices, $A_q = \{i < j < k < l < \ldots\}$.*

a. A monotonically labelled N-tree of order q is a pair of maps $t = (t', t'')$:

$$A_q \setminus \{i\} \xrightarrow{u'} A_q \xrightarrow{u''} \Lambda$$

such that $A(z) < z$ for all $z \in A_q \setminus \{i\}$. The order of t is denoted by $\rho(t) = q$.

b. *The monotonically labelled N-tree of order* o *is denoted by* $\emptyset$.

c. *The set of all monotonically labelled N-trees is denoted by LTN.*

d. *The node with label i is called root.*

The A_q is called number indices set and Λ is called alphabet indices. Any element of LTN can be represented as a graph in which its nodes are the indices for A_q and its edges are the pair $(t'(z), z)$ for $z \in A_q \setminus \{i\}$. The second label of each node is $t''(z)$ for $z \in A_q$.

In Figure 3.1.1 we draw down some element of LTN. The maps $t = (t', t'')$ for the 7th-tree in this figure are given by

- $t' : \{j, k\} \to \{i, j, k\}$ is defined as $t'(j) = i, t'(k) = i$;
- $t'' : \{i, j, k\} \to \{\nu, \mu\}$ is defined as $t''(i) = \nu, t''(j) = \mu$, and $t''(k) = \nu$.

In comparison to the 8th-tree, these two graphs differ from each other only in the labeling of number indices. It means that by rearranging the number indices, these two tree would be identically the same.

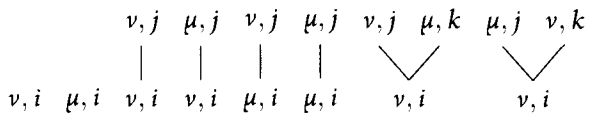


FIGURE 3.1.1: *Sone element of LTN.*

Definition 3.1.2. *Let* $t_1 = (t_1', t_1'')$ *and* $t_2 = (t_2', t_2'')$ *be elements of LTN. An equivalence relation on LTN which is denoted as* $t_1 \sim t_2$ *is defined by:* $u \sim v$ *if and only if*

a. $\rho(t_1) = \rho(t_2) \geqslant 1$;

b. *There exists a permutation* $\sigma : A_q \to A_q$ *such that* $\sigma(i) = i$ *and* $(\sigma \times \sigma)(t) = (\sigma \times \sigma)(t)$ *on* $(A_q \setminus \{i\}) \times A_q$.

Definition 3.1.3. *The set of all equivalence classes under the relation* $\sim$ *is denoted by* $TN = LTN/\sim$. *The elements of TN are called N-trees. The order and the root of a N-tree is defined by the order and the root of a representative, respectively. The notations are* $\rho(t)$ *for the order and* $r(t)$ *for the root. The cardinality of t is denoted by* $a(t)$.

Examples of N-trees are graphically represented in Figure 3.1.2. Next, we discuss recursive representation of N-trees.

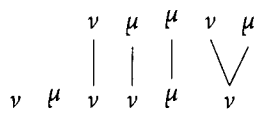


FIGURE 3.1.2: *Examples of N-trees*

Definition 3.1.4. *The $t_1, \ldots, t_m$ be N-trees with non-zero orders and let $v \in \Lambda$. Then we denote*

$$t = [t_1, \ldots, t_m]_v$$

the tree that is obtained by connecting the roots of trees $t_1, \ldots, t_m$ to the new node which is labelled as v. This node becomes the root of the new tree t.

We use notation τ_v for the N-tree of order 1 with root v, for $v \in \Lambda$. For given N-tree of the form $t = [t_1, \ldots, t_m]_v$, its parameters such as order, root, and cardinality are computed with the help from the following lemma.

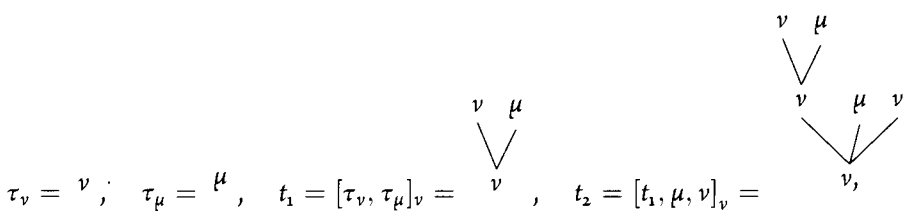


FIGURE 3.1.3: *Some recursive representation of N-trees*

Lemma 3.1.1. *For a given N-tree $t = [t_1, \ldots, t_m]_v$, we have*

$$r(t) = v,$$
$$\rho(t) = 1 + \rho(t_1) + \ldots \rho(t_m),$$

and the cardinality of t is computed recursively as follows:

$$\begin{aligned} a(\emptyset) &= 1, \\ a(\tau_v) &= 1, \quad v \in \Lambda, \\ a(t) &= \binom{\rho(t)-1}{\rho(t_1), \ldots, \rho(t_m)} a(t_1) \ldots a(t_m) \frac{1}{\mu_1! \mu_2! \ldots} \end{aligned}$$

where μ_1, μ_2 are the numbers of mutually equal N-trees among $t_1, \ldots, t_m$.

For trees in Figure 3.1.3 we have

$$\begin{aligned} a(t_1) &= \binom{2}{1,1} \times 1 \times 1 \times \frac{1}{1! \times 1!} = 2, \\ a(t_2) &= \binom{5}{3,1,1} \times 2 \times 1 \times 1 \times \frac{1}{1! \times 1! \times 1!} = 40. \end{aligned}$$

3.1.2 Elementary differentials

We are now apply N-trees to express the Taylor expansion of the function $f : U \to \mathbb{R}^d$, $U \subset \mathbb{R}^d$ which is assumed to be arbitrarily differentiable and be decomposed to several parts, $f = \sum_{v \in \Lambda} f^{[v]}$.

Definition 3.1.5. *For $t \in TN$ with the representation $t = [t_1, \ldots, t_m]_v$ we define a function $F(t) : U \to \mathbb{R}^d$ recursively in the following way:*

$$\begin{aligned} F^i(\emptyset)(y) &= y^i, \\ F^i(\tau_v)(y) &= f^{[v],i}(y), \\ F^i(t)(y) &= \sum_{j_1, \ldots, j_m = 1}^{d} \frac{\partial f^{[v],i}(y)}{\partial y_{j_1} \ldots \partial y_{j_m}} F^{j_1}(t_1)(y) \ldots F^{j_m}(t_m)(y). \end{aligned}$$

The function $F(t)$ is called elementary differentials.

The elementary differential corresponding to the N-tree t_1 in Figure 3.1.3 is

$$F^i(t_1)(y) = \sum_{j,k=1}^{s} f_{j,k}^{[\nu],i} F^j(\tau_\nu)(y) F^k(\tau_\mu)(y)$$
$$= \sum_{j,k=1}^{s} f_{j,k}^{[\nu],i} f^{[\nu],j} f^{[\mu],k}.$$

We now show you how exact solution of (3.1.1) and its derivatives can be expressed by elementary differentials.

Lemma 3.1.2. *Let* $y(t) = \left[y^1(t), \ldots, y^d(t)\right]^T$ *be a solution of* (3.1.1). *We have, for* $q > 0$,

$$\left(y^i\right)^{(q)} = \sum_{\substack{t \in LTN \\ \rho(t)=q}} F^i(t)(y) = \sum_{\substack{t \in TN \\ \rho(t)=q}} a(u) F^i(t)(y). \tag{3.1.3}$$

Proof. We prove the first equality by induction on q. For $t = (t', t'') \in LTN$, $\rho(t) = q$, we have

$$\frac{d}{dy} F^i(t)(y) = \sum_{t_1} F^i(t_1)(y),$$

where the sum is taken over all trees $t_1 \in LTN$ of order $q+1$ such that $t_1'|_{A_q \setminus \{j\}} = t'$ and $t_1''|_{A_q} = t''$. Hence

$$\sum_{\substack{t \in LTN \\ \rho(t)=q}} F^i(t)(y) = \sum_{\substack{t=(t',t'') \in LTN \\ \rho(t)=q}} \sum_{\substack{t_1=(t_1',t_1''), \rho(t_1)=q+1 \\ t_1'|_{A_q \setminus \{j\}} = t', t_1''|_{A_q} = t''}} F^i(t_1)(y)$$
$$= \sum_{\substack{t_1 \in LTN \\ \rho(t_1)=q}} F^i(t_1)(y).$$

The second equality is obvious because of the definition of the $a(t)$. □

Denoting LTN^ν and TN^ν as monotonically labelled N-tree and N-tree with alphabet root ν, the equation (3.1.1) can be rewritten as

$$\left(y^i\right)^{(q)} = \sum_{\nu \in \Lambda} \sum_{\substack{t \in LTN^\nu \\ \rho(t)=q}} F^i(t)(y) = \sum_{\nu \in \Lambda} \sum_{\substack{t \in TN^\nu \\ \rho(t)=q}} a(t) F^i(t)(y). \tag{3.1.4}$$

Using Lemma 3.1.2, the Taylor expansion of a solution of (3.1.1) is given by

$$y(t_0 + \Delta t) = \sum_{q \geqslant 0} \left(\sum_{\nu \in \Lambda} \sum_{\substack{t \in TN^\nu \\ \rho(t) = q}} a(t) F(t) (y_0) \right) \frac{\Delta t^q}{q!} \tag{3.1.5a}$$

$$= \sum_{\nu \in \Lambda} \sum_{t \in TN^\nu} a(t) F(t) (y_0) \frac{\Delta t^{\rho(t)}}{\rho(t)!}. \tag{3.1.5b}$$

3.1.3 N-SERIES

Definition 3.1.6. *Let $f = \sum_{\nu \in \Lambda} f^{[\nu]} : U \to \mathbb{R}^d$ and $y \in U$. A N-series associated with $f^{[\nu]}$ is the series of the form*

$$N^{[\nu],i}(\Phi, y) = \sum_{t \in TN^\nu} \Phi(t) a(t) F^i(t) (y) \frac{\Delta^{\rho(t)}}{\rho(t)!}, \quad i = 1, \ldots, n,$$

where $\Phi : TN \to \mathbb{R}$ is an arbitrary mapping and $\nu \in \Lambda$. The N-series associated with f is of the form

$$N^i(\Phi, y) = \sum_{\nu \in \Lambda} N^{[\nu],i}(\Phi, y).$$

The $\Phi(t)$ are called coefficients.

Definition 3.1.7. *Let $g : (-s_0, s_0) \in \mathbb{R}^d$ be arbitrarily differentiable, $s_0 > 0$ and $g = \left[g^1, \ldots, g^d\right]^T$. We said that g can be represented as a N-series at y w.r.t. f, if and only if there exists a map $\Phi : TN \to \mathbb{R}$ such for all $q \geqslant 0$*

$$\left(g^i\right)^{(q)}(0) = \sum_{\nu \in \Lambda} \sum_{\substack{t \in TN^\nu \\ \rho(t) = q}} \Phi(t) a(t) F^i(t) (y), \quad i = 1, \ldots, n.$$

We denote $g(s) \sim N(\Phi, y)$.

We now reach the central theorem of N-tree theory.

Theorem 3.1.1. *Let $g(s) \sim N(\Phi, y)$ in which $\Phi(\tau_\nu) = 1$ with $\nu \in \Lambda$. Then $\Delta s f^{[\nu]}(g(\Delta s)) \sim$*

$N(\Phi', y)$ *where*

$$\begin{aligned}\Phi'(\emptyset) &= 0,\\ \Phi'(\tau_v) &= 1,\\ \Phi'(t) &= \rho(t)\,\Phi(u_1)\ldots\Phi(t_m),\end{aligned}$$

for $t = [t_1, \ldots, t_m]_v$.

Proof. Firstly, we have to show that

$$\left(\left(f^{[v],i} \circ g\right)(s)\right)^{(q-1)} = \sum_{\substack{t\in S^v\\ \rho(t)=q}} H^i(t)(s) \qquad (3.1.6)$$

in which

$$H^i(t)(s) = \sum_{j_1,\ldots j_m=1}^{n} f^{[v],i}_{j_1,\ldots,j_m}\left(g^{j_1}(s)\right)^{(\rho(u_1))}\ldots\left(g^{j_m}(s)\right)^{(\rho(u_m))}$$

where $u = [u_1, \ldots, u_m]_v$ belongs to S^v, a special class of LTN^v-trees. The set S consists of monotonically labelled N-trees which have no ramification except root and the alphabet indices of the nodes except root are not important. We can formally define S^v as follows

$$S^v = \left\{t = \left(t', t''\right) \in LTN^v \text{ such that card } \left(t'^{-1}\right)(k) \leqslant 1 \text{ for } k \neq j \text{ and } t''(k) = t''(j) \text{ if } k \neq j\right\},$$

and denote S^v_q is the subset of S^v and of order q.

We prove (3.1.6) by induction. For $q = 1$ and 2, we have

$$\begin{aligned}
\left[f^{[v],i}(g(s))\right]^{(1)} &= \sum_{j=1}^{n} f_j^{[v],i}(g(s))\,(g^j)^{(1)}, \\
&= \sum_{t\in S_2^v} H^i(t)(s) \\
\left[f^{[v],i}(g(s))\right]^{(2)} &= \left[\sum_{j=1}^{n} f_j^{[v],i}(g(s))\,(g^j)^{(1)}\right]^{(1)}, \\
&= \sum_{j,k=1}^{n} f_{j,k}^{[v],i}(g^j(s))^{(1)}(g^k(s))^{(1)} + \sum_{j=1}^{n} f_j^{[v],i}(g^j(s))^{(2)} \\
&= \sum_{t\in S_3^v} H^i(t)(s).
\end{aligned}$$

Suppose that

$$\left[f^{[v],i}(g(s))\right]^{q} = \sum_{t\in S_{q+1}^v} H^i(t)(s).$$

Then for $t = [t_1, \ldots, t_m]_v \in S_q^v$, we have

$$\begin{aligned}
\left(H^i(t)(s)\right)' &= \left[f_{j_1,\ldots,j_m}^{v,i}\left[g^{j_1}(s)\right]^{(\rho(t_1))}\cdots\left[g^{j_m}(s)\right]^{(\rho(t_m))}\right]' \\
&= \sum_{j_{m+1}=1}^{n} f_{j_1,\ldots,j_m,j_{m+1}}^{v,i}\left(g^{j_1}(s)\right)^{(\rho(t_1))}\cdots\left(g^{j_m}(s)\right)^{(\rho(t_m))}\left(g^{j_{m+1}}(s)\right)' \\
&\quad + \sum_{k=1}^{m} f_{j_1,\ldots,j_m}^{v,i}\prod_{\substack{l=1\\ l\neq k}}^{m}\left[g^{j_l}(s)\right]^{(\rho(t_l))}\left[g^{j_k}(s)\right]^{(\rho(t_k)+1)}
\end{aligned}$$

The sums of the right hand side are taken all over tree $\tilde{t} \in S_{q+2}^v$ which is received from u in one of two following ways:

i. Add a new node to the root (v, i);

ii. Increase the branch corresponding to the node j_k one more node, $k = 1, \ldots, m$.

Therefore the derivative

$$\left[f^{[v],i}(g(s))\right]^{(q+1)} = \sum_{t\in S_{q+2}^v} H^i(t)(s).$$

Putting $s = 0$, we get

$$\left[f^{[v],i}(g(0))\right]^{(q)} = \sum_{t \in S^v_{q+1}} H^i(t)(0)$$
$$= \sum_{t \in S^v_{q+1}} \sum_{j_1,\ldots,j_m=1}^{n} f^{[v],i}_{j_1,\ldots,j_m} \prod_{k=1}^{m} \left(g^{j_k}(0)\right)^{(\rho(t_k))}.$$

Using the fact that $g(s) \sim N(\Phi, y)$,

$$\left(g^{j_k}(0)\right)^{(\rho(t_k))} = \sum_{\mu_k \in \Lambda} \sum_{\substack{\tilde{t}_k \in LTN^{\mu_k} \\ \rho(\tilde{t}_k) = \rho_{u_k}}} \Phi(v_k) F^{j_k}(\tilde{t}_k)(y),$$

then we return a complicated expression for $\left[f^{[v],i}(g(0))\right]^{(q)}$:

$$\left[f^{[v],i}(g(0))\right]^{(q)} = \sum_{t \in S^v_{q+1}} \prod_{k=1}^{m} \sum_{\mu_k \in \Lambda} \sum_{\substack{\tilde{t}_k \in LTN^{\mu_k} \\ \rho(\tilde{t}_k) = \rho_{u_k}}} \Phi(\tilde{t}_k)$$
$$\times \sum_{j_1,\ldots,j_m=1}^{n} f^{[v],i}_{j_1,\ldots,j_m} F^{j_k}(\tilde{t}_k)(y)$$

The set $\{t, \tilde{t}_1, \ldots, \tilde{t}_m\}$ corresponds to the tree $w \in LTN^v$ such that each branch of t is replaced by the tree $\tilde{t}_1, \ldots, \tilde{t}_m$. Hence

$$\left[f^{[v],i}(g(0))\right]^{(q)} = \sum_{\substack{w \in LTN^v \\ \rho(w) = q+1}} \frac{\Phi'(w)}{\rho(w)} F(w)(y).$$

□

3.2 Rosenbrock explicit Runge-Kutta methods

After spatial discretization, the ADR equations are reduced to an ordinary differential system

$$\frac{du}{dt} = f^{[i]}(u) + f^{[e]}(u) \qquad (3.2.1)$$

in which $f^{[i]}$ is obtained by discretizing $\nabla \cdot F^{\mathrm{vis}} + F^{\mathrm{src}}$ in (2.3.1), $f^{[e]}$ corresponds to the discretization of $-\nabla \cdot F^{\mathrm{inv}}$, and initial value is given.

The properties of these two terms are unfortunately completely different. The term $f^{[i]}$ derived from diffusion and reaction terms is a smooth and stiff operator which contributes real and relatively large scaled eigenvalues as shown in Section 2.2, see also [22, p. 262]. Meanwhile, the term $f^{[e]}$ attained from convection term has predominately imaginary eigenvalues [22, p. 102]. Moreover, $f^{[e]}$ is normally non-smooth because it usually contains itself non-smooth operators such as maximum, minimum functions and limiter functions to avoid spurious oscillations [16] and to preserve positivity property [24].

Because of these differences, temporal integration Runge-Kutta (RK) methods for (3.2.1) are normally of partitioned form. Many authors [4, 30, 38] use explicit RK (ERK) methods for $f^{[e]}$, implicit ones for $f^{[i]}$, and then combine the two methods in proper ways. There are some compromises in these approaches. The implicit methods are often of diagonally implicit RK (DIRK) which offer great stability for stiff term but solving nonlinear large system at every stages is really costly. Alternatives to DIRK is Rosenbrock family of methods, [21, p. 102], [32]. But the associated ERK methods are embedded implicitly in Rosenbrock methods and this situation causes difficulties to find high order of accuracy solvers [39]. Even if ERK methods are accompanied with DIRK directly [26], such combination does not guarantee strong stability-preserving (SSP) property and as a consequence they have to integrate within smaller time step-size.

In this section, we propose a new class of integration methods solving (3.2.1). These methods are affordable in computational cost, having explicit conditions for order of accuracy, and assuring good stability properties. Our approach employes Rosenbrock methods for the stiff term $f^{[i]}$ combining with ERK methods for non-stiff one $f^{[e]}$ directly.

Definition 3.2.1. *An s-stage additive Rosenbrock explicit Runge-Kutta (Ros-ERK) scheme*

solving (3.2.1) *is given by*

$$K_k^{[i]} = \Delta t f^{[i]}\left(u_n + \sum_{j=1}^{k-1} a_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} a_{k,j}^{[e]} K_j^{[e]}\right) + \Delta t J^{[i]}\left(\sum_{j=1}^{k} \beta_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} \beta_{k,j}^{[e]} K_j^{[e]}\right), \quad k = 1, \ldots, s, \tag{3.2.2a}$$

$$K_k^{[e]} = \Delta t f^{[e]}\left(u_n + \sum_{j=1}^{k} \gamma_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} \gamma_{k,j}^{[e]} K_j^{[e]}\right), \quad k = 1, \ldots, s, \tag{3.2.2b}$$

$$u_{n+1} = u_n + \sum_{k=1}^{s} \left(b_k^{[i]} K_k^{[i]} + b_k^{[e]} K_k^{[e]}\right). \tag{3.2.2c}$$

Here, u_n *is approximation of* u *at time* $t = t_n$, $J^{[i]}$ *is the Jacobian of* $f^{[i]}$ *at* t_n *and* $a_{k,j}^{[i]}$, $a_{k,j}^{[e]}$, $\beta_{k,j}^{[i]}$, $\beta_{k,j}^{[e]}$, $\gamma_{k,j}^{[i]}$, $\gamma_{k,j}^{[e]}$, $b_k^{[i]}$ *and* $b_k^{[e]}$ *are determining coefficients.*

Each stage of the method (3.2.2) consists of only a linear system with unknown stage values $K_k^{[i]}$ and with matrices $I - \Delta\beta_{k,k}^{[i]} J^{[i]}$. So, if $\beta_{1,1}^{[i]} = \cdots = \beta_{k,k}^{[i]} = \beta^{[i]}$, there is only one LU-decomposition required per time step.

3.2.1 Order conditions

To specify order conditions for Ros-ERK methods (3.2.2), we use N-trees theory introduced in the previous section. Because the system (3.2.1) consists of two terms, $f^{[i]}$ and $f^{[e]}$, we consider from now only 2-trees with $\Lambda = \{i, e\}$.

Suppose that

$$K_k^{[i]} \sim N^{[i]}\left(\Phi_k, y_n\right), \quad K_k^{[e]} \sim N^{[e]}\left(\Phi_k, y_n\right).$$

It means that for $q \geqslant 0$ we have

$$\left(K_k^{[i]}\right)^{(q)}\Big|_{\Delta t=0} = \sum_{u \in \mathrm{TN}_q^{[i]}} \Phi_k(u)\, a(u)\, F(u)\left(y_n\right),$$

$$\left(K_k^{[e]}\right)^{(q)}\Big|_{\Delta t=0} = \sum_{u \in \mathrm{TN}_q^{[e]}} \Phi_k(u)\, a(u)\, F(u)\left(y_n\right),$$

where the coefficient Φ_k satisfies

$$\Phi_k(\emptyset) = 0 \quad \text{and} \quad \Phi_k(\tau_i) = \Phi_k(\tau_e) = 1.$$

Then,

$$\begin{aligned}
y_n + \sum_{j=1}^{k-1} a_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} a_{k,j}^{[e]} K_j^{[e]} &\sim N\left(\widetilde{\Psi}_k, y_n\right), \\
y_n + \sum_{j=1}^{k} \gamma_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} \gamma_{k,j}^{[e]} K_j^{[e]} &\sim N\left(\widehat{\Psi}_k, y_n\right), \\
\sum_{j=1}^{k} \beta_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} \beta_{k,j}^{[e]} K_j^{[e]} &\sim N\left(\overline{\Psi}_k, y_n\right),
\end{aligned}$$

in which

$$\widetilde{\Psi}_k(u) = \begin{cases} 1 & \text{if } u = \emptyset, \\ \sum_{j=1}^{k-1} a_{k,j}^{[r(u)]} \Phi_j(u) & \text{if } \rho(u) \geqslant 1, \end{cases}$$

$$\widehat{\Psi}_k(u) = \begin{cases} 1 & \text{if } u = \emptyset, \\ \sum_{j=1}^{k} \gamma_{k,j}^{[i]} \Phi_j(u) & \text{if } \rho(u) \geqslant 1,\ r(u) = i, \\ \sum_{j=1}^{k-1} \gamma_{k,j}^{[e]} \Phi_j(u) & \text{if } \rho(u) \geqslant 1,\ r(u) = e, \end{cases}$$

$$\overline{\Psi}_k(u) = \begin{cases} 0 & \text{if } u = \emptyset, \\ \sum_{j=1}^{k} \beta_{k,j}^{[i]} \Phi_j(u) & \text{if } \rho(u) \geqslant 1,\ r(u) = i, \\ \sum_{j=1}^{k-1} \beta_{k,j}^{[e]} \Phi_j(u) & \text{if } \rho(u) \geqslant 1,\ r(u) = e. \end{cases}$$

Applying Theorem 3.1.1, we obtain

$$\begin{aligned}
\Delta t f^{[i]}\left(y_n + \sum_{j=1}^{k} a_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} a_{k,j}^{[e]} K_j^{[e]}\right) &\sim N^{[i]}\left(\widehat{\Psi}_k', y_n\right) \\
\Delta t f^{[e]}\left(y_n + \sum_{j=1}^{k} \gamma_{k,j}^{[i]} K_j^{[i]} + \sum_{j=1}^{k-1} \gamma_{k,j}^{[e]} K_j^{[e]}\right) &\sim N^{[e]}\left(\widetilde{\Psi}_k', y_n\right),
\end{aligned}$$

in which

$$\widetilde{\Psi}'_k(u) = \begin{cases} 0 & \text{if } u = \emptyset, \\ 1 & \text{if } u = \tau_e, \\ \rho(u)\,\widetilde{\Psi}_k(u_1)\ldots\widetilde{\Psi}_k(u_m) & \text{if } u = [u_1,\ldots,u_m]_i, \end{cases}$$

$$\widehat{\Psi}'_k(u) = \begin{cases} 0 & \text{if } u = \emptyset, \\ 1 & \text{if } u = \tau_e, \\ \rho(u)\,\widehat{\Psi}_k(u_1)\ldots\widehat{\Psi}_k(u_m) & \text{if } u = [u_1,\ldots,u_m]_e. \end{cases}$$

$$\Delta t J^{[i]}\left(\sum_{j=1}^{k}\beta^{[i]}_{k,j}K^{[i]}_j + \sum_{j=1}^{k-1}\beta^{[e]}_{k,j}K^{[e]}_j\right) \sim N\left(\overline{\Psi}'_k, y_n\right)$$

with

$$\overline{\Psi}'_k(u) = \begin{cases} \rho(u)\,\overline{\Psi}_k(v) & \text{if } u = [v]_i,\ \rho(v) \geqslant 1, \\ 0 & \text{otherwise.} \end{cases}$$

Gathering all together we come to

$$\text{Left hand side of (3.2.2a)} \sim N^{[i]}\left(\Xi_k, y_n\right),$$
$$\text{Left hand side of (3.2.2b)} \sim N^{[e]}\left(\Xi_k, y_n\right),$$

in which if $r(u) = i$,

$$\Xi_k(u) = \begin{cases} 0 & \text{if } u = 0, \\ 1 & \text{if } u = \tau^{[i]}, \\ \rho(u)\left[\widetilde{\Psi}_k(v) + \overline{\Psi}_k(v)\right] & \text{if } u = [v]_i,\ \rho(v) \geqslant 1, \\ \rho(u)\,\widetilde{\Psi}_k(u_1)\ldots\widetilde{\Psi}_k(u_m) & \text{if } u = [u_1,\ldots,u_m]_i,\ m \geqslant 2, \end{cases}$$

and $\Xi_k(u) = \widehat{\Psi}'_k(u)$ if $r(u) = e$.

Because Ξ_k and Φ_k are identical, we conclude that the coefficient Φ_k are defined as follow

- $\Phi_k(u) = 0$ if $u = \emptyset$;
- $\Phi_k(u) = 1$ if $u \in \left\{\tau^{[e]}, \tau^{[i]}\right\}$;

- For $\rho(u) \geqslant 2$ and $r(u) = e$ then

$$\Phi_k(u) = \rho(u) \prod_{l=1}^{m} \begin{cases} \sum_{j=1}^{k} \gamma_{k,j}^{[i]} \Phi_j(u_l), & \text{if } r(u_l) = i, \\ \sum_{j=1}^{k-1} \gamma_{k,j}^{[e]} \Phi_j(u_l), & \text{if } r(u_l) = e, \end{cases}$$

with $u = [u_1, \ldots, u_m]_e$ and $m \geqslant 1$;

- For $\rho(u) \geqslant 2$ and $r(u) = i$ then

  - If $u = [v]_i$ and $\rho(v) \geqslant 1$ then

$$\Phi_k(u) = \begin{cases} \rho(u) \sum_{j=1}^{k-1} \left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]} \right) \Phi_j(v) & \text{if } r(v) = e, \\ \rho(u) \left[\sum_{j=1}^{k-1} \left(a_{k,j}^{[i]} + \beta_{k,j}^{[i]} \right) \Phi_j(v) + \beta_{k,k}^{[i]} \Phi_k(v) \right] & \text{if } r(v) = i; \end{cases}$$

  - If $u = [u_1, \ldots, u_m]_i$ and $m \geqslant 2$ then

$$\Phi_k(u) = \rho(u) \prod_{l=1}^{m} \sum_{j=1}^{k-1} a_{k,j}^{[r(u_l)]} \Phi_j(u_l).$$

For the numerical solution we have

$$y_{n+1} \sim N(\Phi, y_n)$$

where

$$\Phi(u) = \begin{cases} 1 & \text{if } \rho(u) = 0, \\ \sum_{k=1}^{s} b_k^{[r(u)]} \Phi_k(u) & \text{if } \rho(u) \geqslant 1. \end{cases}$$

Comparing the N-series of the exact solution, we are able to derive the local truncation error of the Ros-ERK method (3.2.2) as

$$y(t_{n+1}) - y_{n+1} = \sum_{v \in \{i,e\}} \sum_{u \in \mathrm{TN}^{[v]}} (1 - \Phi(u)) \, a(u) \, F(u)(y_n) \, \frac{\Delta t^{\rho(u)}}{\rho(u)!}.$$

Therefore, order conditions for Ros-ERK method (3.2.2) to be of order p are determined in the following proposition.

Proposition 3.2.1. *The Ros-ERK method* (3.2.2) *solving* (3.2.1) *is of order p if and only*

if it holds true that

$$\Phi(t) = 1 \textit{ for all } t \in \mathrm{TN} \textit{ satisfying } \rho(t) \leqslant p.$$

EXPLICIT CONDITIONS UP TO THIRD ORDER OF ACCURACY

Here we give explicit formulas of the order conditions up to the third order of accuracy.

Conditions for the first order

- $u^{[e]}_{1,1} = \tau_e$:

$$\Phi_k\left(u^{[e]}_{1,1}\right) = 1 \implies \sum_{k=1}^{s} b^{[e]}_k = 1;$$

- $u^{[i]}_{1,1} = \tau_i$:

$$\Phi_k\left(u^{[i]}_{1,1}\right) = 1 \implies \sum_{k=1}^{s} b^{[i]}_k = 1.$$

Conditions for the second order

- $u^{[e]}_{2,1} = \left[u^{[e]}_{1,1}\right]_e$:

$$\Phi_k\left(u^{[e]}_{2,1}\right) = 2\sum_{j=1}^{k-1} \gamma^{[e]}_{k,j}\Phi_j\left(u^{[e]}_{1,1}\right) = 2\sum_{j=1}^{k-1} \gamma^{[e]}_{k,j},$$

$$\implies 2\sum_{k=1}^{s} b^{[e]}_k \sum_{j=1}^{k-1} \gamma^{[e]}_{k,j} = 1;$$

- $u^{[e]}_{2,2} = \left[u^{[i]}_{1,1}\right]_e$:

$$\Phi_k\left(u^{[e]}_{2,2}\right) = 2\sum_{j=1}^{k} \gamma^{[i]}_{k,j}\Phi_j\left(u^{[i]}_{1,1}\right) = 2\sum_{j=1}^{k} \gamma^{[i]}_{k,j},$$

$$\implies 2\sum_{k=1}^{s} b^{[e]}_k \sum_{j=1}^{k} \gamma^{[i]}_{k,j} = 1;$$

- $u_{2,1}^{[i]} = \left[u_{1,1}^{[e]}\right]_i$:

$$\Phi_k\left(u_{2,1}^{[i]}\right) = 2\sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\Phi_j\left(u_{1,1}^{[e]}\right) = 2\sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right),$$

$$\Longrightarrow 2\sum_{k=1}^{s} b_k^{[i]} \sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right) = 1;$$

- $u_{2,2}^{[i]} = \left[u_{1,1}^{[i]}\right]_i$:

$$\Phi_k\left(u_{2,2}^{[i]}\right) = 2\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]} + \beta_{k,j}^{[i]}\right)\Phi_j\left(u_{1,1}^{[i]}\right) + \beta_{k,k}^{[i]}\Phi_j\left(u_{1,1}^{[i]}\right)\right]$$

$$= 2\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]} + \beta_{k,j}^{[i]}\right) + \beta_{k,j}^{[i]}\right],$$

$$\Longrightarrow 2\sum_{k=1}^{s} b_k^{[i]}\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]} + \beta_{k,j}^{[i]}\right) + \beta_{k,j}^{[i]}\right] = 1.$$

Conditions for the third order

- $u_{3,1}^{[e]} = \left[u_{1,1}^{[e]}, u_{1,1}^{[e]}\right]_e$:

$$\Phi_k\left(u_{3,1}^{[e]}\right) = 3\left[\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\Phi_j\left(u_{1,1}^{[e]}\right)\right]^2 = 3\left(\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\right)^2,$$

$$\Longrightarrow 3\sum_{k=1}^{s} b_k^{[e]}\left(\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\right)^2 = 1;$$

- $u_{3,2}^{[e]} = \left[u_{1,1}^{[e]}, u_{1,1}^{[i]}\right]_e$:

$$\Phi_k\left(u_{3,2}^{[e]}\right) = 3\left(\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\Phi_j\left(u_{1,1}^{[e]}\right)\right)\left(\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\Phi_j\left(u_{1,1}^{[i]}\right)\right)$$

$$= 3\left(\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\right)\left(\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\right)$$

$$\Longrightarrow 3\sum_{k=1}^{s} b_k^{[e]}\left(\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\right)\left(\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\right) = 1;$$

- $u_{3,3}^{[e]} = \left[u_{1,1}^{[i]}, u_{1,1}^{[i]}\right]_e$:

$$\Phi_k\left(u_{3,3}^{[e]}\right) = 3\left(\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\Phi_j\left(u_{1,1}^{[i]}\right)\right)^2 = 3\left(\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\right)^2$$

$$\Longrightarrow 3\sum_{k=1}^{s} b_k^{[e]}\left(\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\right)^2 = 1;$$

- $u_{3,4}^{[e]} = \left[u_{2,1}^{[e]}\right]_e$:

$$\Phi_k\left(u_{3,4}^{[e]}\right) = 3\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\Phi_j\left(u_{2,1}^{[e]}\right) = 6\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\sum_{l=1}^{j-1}\gamma_{j,l}^{[e]}$$

$$\Longrightarrow 6\sum_{k=1}^{s} b_k\left[\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\left(\sum_{l=1}^{j-1}\gamma_{j,l}^{[e]}\right)\right] = 1;$$

- $u_{3,5}^{[e]} = \left[u_{2,2}^{[e]}\right]_e$:

$$\Phi_k\left(u_{3,5}^{[e]}\right) = 3\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\Phi_j\left(u_{2,2}^{[e]}\right) = 6\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\left(\sum_{l=1}^{j}\gamma_{j,l}^{[i]}\right)$$

$$\Longrightarrow 6\sum_{k=1}^{s} b_k\left[\sum_{j=1}^{k-1}\gamma_{k,j}^{[e]}\left(\sum_{l=1}^{j}\gamma_{j,l}^{[i]}\right)\right] = 1;$$

- $u_{3,6}^{[e]} = \left[u_{2,1}^{[i]}\right]_e$:

$$\Phi_k\left(u_{3,6}^{[e]}\right) = 3\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\Phi_j\left(u_{2,1}^{[i]}\right) = 6\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\left[\sum_{l=1}^{j-1}\left(a_{j,l}^{[e]} + \beta_{j,l}^{[e]}\right)\right]$$

$$\Longrightarrow 6\sum_{k=1}^{s} b_k^{[e]}\left\{\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\left[\sum_{l=1}^{j-1}\left(a_{j,l}^{[e]} + \beta_{j,l}^{[e]}\right)\right]\right\} = 1;$$

- $u_{3,7}^{[e]} = \left[u_{2,2}^{[i]}\right]_i$:

$$\Phi_k\left(u_{3,7}^{[e]}\right) = 3\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\Phi_j\left(u_{2,1}^{[i]}\right) = 6\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\left[\sum_{l=1}^{j-1}\left(a_{j,l}^{[i]} + \beta_{j,l}^{[i]}\right) + \beta_{j,j}^{[i]}\right]$$

$$\Longrightarrow 6\sum_{k=1}^{s} b_k^{[e]}\left\{\sum_{j=1}^{k}\gamma_{k,j}^{[i]}\left[\sum_{l=1}^{j-1}\left(a_{j,l}^{[i]} + \beta_{j,l}^{[i]}\right) + \beta_{j,j}^{[i]}\right]\right\} = 1;$$

- $u_{3,1}^{[i]} = \left[u_{1,1}^{[e]}, u_{1,1}^{[e]}\right]_i$:

$$\Phi_k\left(u_{3,1}^{[i]}\right) = 3\left(\sum_{j=1}^{k-1} a_{k,j}^{[e]}\Phi_j\left(u_{1,1}^{[e]}\right)\right)^2 = 3\left(\sum_{j=1}^{k-1} a_{k,j}^{[e]}\right)^2$$

$$\Longrightarrow 3\sum_{k=1}^{s} b_k^{[i]}\left(\sum_{j=1}^{k-1} a_{k,j}^{[e]}\right)^2 = 1;$$

- $u_{3,2}^{[i]} = \left[u_{1,1}^{[e]}, u_{1,1}^{[i]}\right]_i$:

$$\Phi_k\left(u_{3,2}^{[i]}\right) = 3\left(\sum_{j=1}^{k-1} a_{k,j}^{[e]}\right)\left(\sum_{j=1}^{k-1} a_{k,j}^{[i]}\right)$$

$$\Longrightarrow 3\sum_{k=1}^{s} b_k^{[i]}\left(\sum_{j=1}^{k-1} a_{k,j}^{[e]}\right)\left(\sum_{j=1}^{k-1} a_{k,j}^{[i]}\right) = 1;$$

- $u_{3,3}^{[i]} = \left[u_{1,1}^{[i]}, u_{1,1}^{[i]}\right]_i$:

$$\Phi_k\left(u_{3,3}^{[i]}\right) = 3\left(\sum_{j=1}^{k-1} a_{k,j}^{[i]}\right)^2$$

$$\Longrightarrow 3\sum_{k=1}^{s} b_k^{[i]}\left(\sum_{j=1}^{k-1} a_{k,j}^{[i]}\right)^2 = 1;$$

- $u_{3,4}^{[i]} = \left[u_{2,1}^{[e]}\right]_i$:

$$\Phi_k\left(u_{3,4}^{[i]}\right) = 3\sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\Phi_j\left(u_{2,1}^{[e]}\right) = 6\sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\left(\sum_{l=1}^{j-1}\gamma_{j,l}^{[e]}\right)$$

$$\Longrightarrow 6\sum_{k=1}^{s} b_k^{[i]}\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\left(\sum_{l=1}^{j-1}\gamma_{j,l}^{[e]}\right)\right] = 1;$$

- $u_{3,5}^{[i]} = \left[u_{2,2}^{[e]}\right]_i$:

$$\Phi_k\left(u_{3,5}^{[i]}\right) = 3\sum_{j=1}^{k-1}\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\Phi_j\left(u_{2,2}^{[e]}\right) = 6\sum_{j=1}^{k-1}\left[\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\left(\sum_{l=1}^{j}\gamma_{k,j}^{[i]}\right)\right]$$

$$\Longrightarrow 6\sum_{k=1}^{s} b_k^{[i]}\left\{\sum_{j=1}^{k-1}\left[\left(a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right)\left(\sum_{l=1}^{j}\gamma_{k,j}^{[i]}\right)\right]\right\} = 1;$$

- $u_{3,6}^{[i]} = \left[u_{2,1}^{[i]}\right]_i$:

$$\begin{aligned}
\Phi_k\left(u_{3,6}^{[i]}\right) &= 3\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]}+\beta_{k,j}^{[i]}\right)\Phi_j\left(u_{2,1}^{[i]}\right)+\beta_{k,k}^{[i]}\Phi_k\left(u_{2,1}^{[i]}\right)\right] \\
&= 6\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]}+\beta_{k,j}^{[i]}\right)\sum_{l=1}^{j-1}\left(a_{j,l}^{[e]}+\beta_{j,l}^{[e]}\right)+\beta_{k,k}^{[i]}\sum_{l=1}^{k-1}\left(a_{k,l}^{[e]}+\beta_{k,l}^{[e]}\right)\right] \\
\Longrightarrow 6\sum_{k=1}^{s} b_k^{[i]}&\left[\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]}+\beta_{k,j}^{[i]}\right)\sum_{l=1}^{j-1}\left(a_{j,l}^{[e]}+\beta_{j,l}^{[e]}\right)+\beta_{k,k}^{[i]}\sum_{l=1}^{k-1}\left(a_{k,l}^{[e]}+\beta_{k,l}^{[e]}\right)\right] \\
&= 1;
\end{aligned}$$

- $u_{3,7}^{[i]} = \left[u_{2,2}^{[i]}\right]_i$:

$$\begin{aligned}
\Phi\left(u_{3,7}^{[i]}\right) &= 6\left\{\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]}+\beta_{k,j}^{[i]}\right)\left[\sum_{l=1}^{j-1}\left(a_{j,l}^{[i]}+\beta_{j,l}^{[i]}\right)+\beta_{j,j}^{[i]}\right]\right. \\
&\qquad \left.+\beta_{k,k}^{[i]}\left[\sum_{l=1}^{k-1}\left(a_{k,l}^{[i]}+\beta_{k,l}^{[i]}\right)+\beta_{k,k}^{[i]}\right]\right\} \\
\Longrightarrow 6\sum_{k=1}^{s} b_k^{[i]}&\left\{\sum_{j=1}^{k-1}\left(a_{k,j}^{[i]}+\beta_{k,j}^{[i]}\right)\left[\sum_{l=1}^{j-1}\left(a_{j,l}^{[i]}+\beta_{j,l}^{[i]}\right)+\beta_{j,j}^{[i]}\right]\right. \\
&\qquad \left.+\beta_{k,k}^{[i]}\left[\sum_{l=1}^{k-1}\left(a_{k,l}^{[i]}+\beta_{k,l}^{[i]}\right)+\beta_{k,k}^{[i]}\right]\right\} = 1.
\end{aligned}$$

3.2.2 Stability function

The linear stability function for an s-stage Ros-ERK method is obtained by applying it to the linear test equation

$$\frac{dy}{dt} = \lambda^{[i]}y + \lambda^{[e]}y, \tag{3.2.3}$$

and getting the relation

$$\begin{bmatrix} I - z^{[i]}C^{[i]} & -z^{[e]}C^{[e]} & o \\ -z^{[i]}G^{[i]} & I - z^{[e]}G^{[e]} & o \\ -\left(b^{[i]}\right)^T & -\left(b^{[e]}\right)^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K}^{[i]} \\ \mathbf{K}^{[e]} \\ y_{n+1} \end{bmatrix} = y_n \begin{bmatrix} z^{[i]}1 \\ z^{[e]}1 \\ 1 \end{bmatrix},$$

where $I \in \mathbb{R}^{s\times s}$ is the identity matrix, $z^{[i]} = \lambda^{[i]}\Delta t, z^{[e]} = \lambda^{[e]}\Delta t, 1 = [1, \ldots, 1]^T \in \mathbb{R}^{s\times 1}$, $\mathbf{K}^{[i]} = \left[K_1^{[i]}, \ldots, K_s^{[i]}\right]^T, \mathbf{K}^{[e]} = \left[K_1^{[e]}, \ldots, K_s^{[e]}\right]^T$, and

$$C^{[i]} = \left[a_{k,j}^{[i]} + \beta_{k,j}^{[i]}\right] \in \mathbb{R}^{s\times s}, \qquad C^{[e]} = \left[a_{k,j}^{[e]} + \beta_{k,j}^{[e]}\right] \in \mathbb{R}^{s\times s},$$
$$G^{[i]} = \left[\gamma_{k,j}^{[i]}\right] \in \mathbb{R}^{s\times s}, \qquad G^{[e]} = \left[\gamma_{k,j}^{[e]}\right] \in \mathbb{R}^{s\times s}.$$

Then the stability function for the Ros-ERK method (3.2.2) is given by

$$R\left(z^{[i]}, z^{[e]}\right) = \frac{P\left(z^{[i]}, z^{[e]}\right)}{Q\left(z^{[i]}, z^{[e]}\right)}, \tag{3.2.4}$$

in which

$$P\left(z^{[i]}, z^{[e]}\right) = \det \begin{bmatrix} I - z^{[i]}\left[C^{[i]} - 1\left(b^{[i]}\right)^T\right] & -z^{[i]}C^{[e]} + z^{[i]}1\left(b^{[e]}\right)^T \\ -z^{[e]}G^{[i]} + z^{[e]}1\left(b^{[i]}\right)^T & I - z^{[e]}\left[G^{[e]} - 1\left(b^{[e]}\right)^T\right] \end{bmatrix},$$
$$Q\left(z^{[i]}, z^{[e]}\right) = \det \begin{bmatrix} I - z^{[i]}C^{[i]} & -z^{[i]}C^{[e]} \\ -z^{[e]}G^{[i]} & I - z^{[e]}G^{[e]} \end{bmatrix}$$

If coefficients of Ros-ERK methods satisfy the additional restrictions

$$G^{[e]} = C^{[e]}, \quad G^{[i]} = C^{[i]},$$

then the function $R\left(z^{[i]}, z^{[e]}\right)$ reduces to

$$R\left(z^{[i]}, z^{[e]}\right) = \frac{\det\left(I - z^{[i]}C^{[i]} - z^{[e]}C^{[e]} + z^{[i]}1\left(b^{[i]}\right)^T + z^{[e]}1\left(b^{[e]}\right)^T\right)}{\det\left(I - z^{[i]}C^{[i]} - z^{[e]}C^{[e]}\right)}.$$

It should be noticed that the matrix $z^{[i]}C^{[i]} + z^{[e]}C^{[e]}$ is triangular with diagonal entries

$z^{[i]}\beta_{k,k}^{[i]}$, for $k = 1, \ldots, s$. Therefore, the denominator is

$$Q\left(z^{[i]}, z^{[e]}\right) = \prod_{k=1}^{s}\left(1 - z^{[i]}\beta_{k,k}^{[i]}\right).$$

In the case of $\beta_{k,k}^{[i]} \equiv \beta^{[i]}$ for all $k = 1, \ldots, s$, we have $Q\left(z^{[i]}, z^{[e]}\right) = \left(1 - z^{[i]}\beta^{[i]}\right)^{s}$.

3.2.3 Examples on Rosenbrock explicit Runge-Kutta methods

Two examples belonging to the Ros-ERK methods are introduced in this section. They are all s-stage, $s - 1$ order of accuracy and denoted by Ros-ERKs, $s - 1$ for $s = 2$ and 3.

The first one is Ros-ERK2, 1 which is given by

$$\left[a_{k,j}^{[i]}\right] = \left[a_{k,j}^{[e]}\right] = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & 0 \end{bmatrix}, \quad \left[\beta_{k,j}^{[i]}\right] = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad \left[\beta_{k,j}^{[e]}\right] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$
$$\left[\gamma_{k,j}^{[i]}\right] = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad \left[\gamma_{k,j}^{[e]}\right] = \begin{bmatrix} 0 & 0 \\ \frac{1}{2} & 0 \end{bmatrix}, \quad \left[b_k^{[i]}\right] = \left[b_k^{[e]}\right] = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Its stability function is

$$R_{\text{Ros-ERK2,1}}\left(z^{[i]}, z^{[e]}\right) = \frac{\left(1 + \frac{1}{2}z^{[e]}\right)^2}{\left(1 - \frac{1}{2}z^{[i]}\right)^2}.$$

The next one is Ros-ERK3, 2 which is defined by

$$\left[a_{k,j}^{[i]}\right] = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{6} & 0 & 0 \\ \frac{1}{4} & \frac{1}{12} & 0 \end{bmatrix}, \quad \left[a_{k,j}^{[e]}\right] = \left[\beta_{k,j}^{[e]}\right] = \frac{1}{2}\left[\gamma_{k,j}^{[e]}\right] = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \end{bmatrix},$$
$$\left[\beta_{k,j}^{[i]}\right] = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{4} & \frac{1}{12} & \frac{1}{3} \end{bmatrix}, \quad \left[\gamma_{k,j}^{[i]}\right] = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} \end{bmatrix}, \quad \left[b_k^{[i]}\right] = \left[b_k^{[e]}\right] = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{6} \\ \frac{1}{3} \end{bmatrix}.$$

The stability function of this method is

$$R_{\text{Ros-ERK3,2}}\left(z^{[i]}, z^{[e]}\right) = \frac{1 + z^{[e]} + \frac{1}{2}\left(z^{[e]}\right)^2 + \frac{1}{6}\left(z^{[e]}\right)^3 - \left(\frac{1}{6} - \frac{7}{54}z^{[e]}\right)\left(z^{[i]}\right)^2}{\left(1 - \frac{1}{3}z^{[i]}\right)^3}.$$

It is impossible to require that these methods are L-stable for all value of $\lambda^{[e]}$. But under the assumption that $z^{[e]} = \mathcal{O}(1)$, the stability functions tend to zero as $z^{[i]}$ tends to $-\infty$. This damping property at infinity assures the L-stability of our methods for large and negative eigenvalues of stiff term.

3.3 Additive Rosenbrock strong stability-preserving methods

As mentioned before, the $f^{[e]}$ is resulted from a method of lines approximation of the advection part $-\nabla \cdot F^{\text{inv}}$ where the spatial derivative is discretized by a total variational diminishing (TVD) finite difference or finite element approximation. To solve the system

$$\frac{du}{dt} = f^{[e]}(u), \tag{3.3.1}$$

the strong stability-preserving (SSP) methods is preferred [19] because they do not increase the computational cost so much and have the extra assurance of stability.

To employ SSP methods for integrating the term $f^{[e]}$ in the full system (3.2.1), the formulas (3.2.2b) and (3.2.2c) must be rewritten as

$$Y_k = u_n + \sum_{j=1}^{k} \zeta_{k,j}^{[i]} K_j^{[i]} + \Delta t \sum_{j=1}^{k-1} \zeta_{k,j}^{[e]} f^{[e]}(Y_j), \quad k = 1, \ldots, s+1,$$

$$u_{n+1} = Y_{s+1},$$

respectively, in which

$$\left[\zeta_{k,j}^{[i]}\right] = \begin{bmatrix} \gamma_{1,1}^{[i]} & 0 & \ldots & 0 & 0 \\ \gamma_{2,1}^{[i]} & \gamma_{2,2}^{[i]} & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{s,1}^{[i]} & \gamma_{s,2}^{[i]} & \ldots & \gamma_{s,s}^{[i]} & 0 \\ b_1^{[i]} & b_2^{[i]} & \ldots & b_s^{[i]} & 0 \end{bmatrix}, \quad \left[\zeta_{k,j}^{[e]}\right] = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ \gamma_{2,1}^{[e]} & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{s,1}^{[e]} & \gamma_{s,2}^{[e]} & \ldots & 0 & 0 \\ b_1^{[e]} & b_2^{[e]} & \ldots & b_s^{[e]} & 0 \end{bmatrix}.$$

Notice that the value $K_{s+1}^{[i]}$ is not required because of $\zeta_{s+1,s+1}^{[i]} = 0$. For each stage $k \geqslant 2$

with given $\eta_{k,j} \geqslant 0$ satisfying $\sum_{j=1}^{k-1} \eta_{k,j} = 1$, we have

$$\begin{aligned} Y_k &= \sum_{j=1}^{k-1} \eta_{k,j} \left[Y_j - \sum_{l=1}^{j} \zeta_{j,l}^{[i]} K_l^{[i]} - \Delta t \sum_{l=1}^{j-1} \zeta_{j,l}^{[e]} f^{[e]}(Y_l) \right] \\ &\quad + \sum_{j=1}^{k} \zeta_{k,j}^{[i]} K_j^{[i]} + \Delta t \sum_{j=1}^{k-1} \zeta_{k,j}^{[e]} f^{[e]}(Y_j) \\ &= \sum_{j=1}^{k-1} \left[\eta_{k,j} Y_j + \Delta t \left(\zeta_{k,j}^{[e]} - \sum_{l=j+1}^{k-1} \eta_{k,l} \zeta_{l,j}^{[e]} \right) f^{[e]}(Y_j) \right] \\ &\quad + \sum_{j=1}^{k} \left(\zeta_{k,j}^{[i]} - \sum_{l=j}^{k-1} \eta_{k,j} \zeta_{l,j}^{[i]} \right) K_j^{[i]}. \end{aligned}$$

So, if we denote

$$\theta_{k,j}^{[i]} = \zeta_{k,j}^{[i]} - \sum_{l=j}^{k-1} \eta_{k,j} \zeta_{l,j}^{[i]}, \quad \theta_{k,j}^{[e]} = \zeta_{k,j}^{[e]} - \sum_{l=j+1}^{k-1} \eta_{k,l} \zeta_{l,j}^{[e]},$$

(3.2.2) is written in the equivalent form

$$\begin{aligned} K_k^{[i]} &= \Delta t f^{[i]} \left(u_n + \sum_{j=1}^{k-1} a_{k,j}^{[i]} K_j^{[i]} + \Delta t \sum_{j=1}^{k-1} a_{k,j}^{[e]} f^{[e]}(Y_j) \right) \\ &\quad + \Delta t J^{[i]} \left(\sum_{j=1}^{k} \beta_{k,j}^{[i]} K_j^{[i]} + \Delta t \sum_{j=1}^{k-1} \beta_{k,j}^{[e]} f^{[e]}(Y_j) \right), \quad k = 1, \ldots, s, \end{aligned} \tag{3.3.2a}$$

$$Y_1 = u_n + \theta_{1,1}^{[i]} K_1^{[i]}, \tag{3.3.2b}$$

$$Y_k = \sum_{j=1}^{k-1} \left(\eta_{k,j} Y_j + \Delta t \theta_{k,j}^{[e]} f^{[e]}(Y_j) \right) + \sum_{j=1}^{k} \theta_{k,j}^{[i]} K_j^{[i]}, \quad k = 2, \ldots, s+1, \tag{3.3.2c}$$

$$u_{n+1} = Y_{s+1}. \tag{3.3.2d}$$

Under some assumptions mentioned later, this scheme would be called Rosenbrock strongly stability-preserving (Ros-SSP) methods.

Clearly, if all $\theta_{k,j}^{[e]}$'s are nonnegative, the intermediate stages Y_k consist of convex combinations of forward Euler operators for $f^{[e]}$ and ones of stage values for $f^{[i]}$. Then we

assume that the forward Euler method applied to (3.3.1), i.e.,

$$u_{n+1} = u_n + \Delta t f^{[e]}(u_n), \tag{3.3.3}$$

is strongly stable, $\left\| u_n + \Delta t f^{[e]}(u_n) \right\| \leqslant \|u_n\|$ with $\Delta t \leqslant \Delta t_{\mathrm{FE}}$, and the stage values K_k are also strongly stable, $\left\| K_k^{[i]} \right\| \leqslant \|u_n\|$ with $\Delta t \leqslant \Delta t_{\mathrm{Ros}}$. Obviously, we get that u_{n+1} obtained by (3.3.2) is stable with

$$\Delta t \leqslant \min \left\{ \Delta t_{\mathrm{FE}} \min_{k,j} \frac{\eta_{k,j}}{\theta_{k,j}^{[e]}}, \Delta t_{\mathrm{Ros}} \frac{1}{\max_k \sum_{j=1}^{k} \left| \theta_{k,j}^{[i]} \right|} \right\}.$$

In case there are some negative $\theta_{k,j}^{[e]}$, we will use the same trick as in [34] in which an operator so called backward operator $\widetilde{f}^{[e]}$ is introduced. This can be achieved by spatially discretizing the temporal reversed system

$$\frac{\partial u}{\partial t} = \nabla \cdot F^{\mathrm{inv}}(u).$$

The operator $\widetilde{f}^{[e]}$ is required to hold the strong stability property, $\|u_{n+1}\| \leqslant \|u_n\|$, for the backward Euler scheme,

$$u_{n+1} = u_n - \Delta t \widetilde{f}^{[e]}(u_n). \tag{3.3.4}$$

Then we have the following proposition which may be considered as an extension of that in [34].

Proposition 3.3.1. *Assume that the forward Euler* (3.3.3) *and the backward Euler* (3.3.4) *are strongly stable, i.e.,*

$$\left\| u_n + \Delta t f^{[e]}(u_n) \right\| \leqslant \|u_n\| \text{ and } \left\| u_n - \Delta t \widetilde{f}^{[e]}(u_n) \right\| \leqslant \|u_n\|,$$

with $\Delta t \leqslant \Delta t_{\mathrm{FE}}$, *and stage values* $K_k^{[i]}$ *are strongly stable with* $\Delta t \leqslant \Delta t_{\mathrm{Ros}}$. *Then the method* (3.3.2) *is strongly stable-preserving,* $\|u_{n+1}\| \leqslant \|u_n\|$, *with*

$$\Delta t \leqslant \min \left\{ \Delta t_{\mathrm{FE}} \min_{k,j} \frac{\eta_{k,j}}{\left| \theta_{k,j}^{[e]} \right|}, \Delta t_{\mathrm{Ros}} \frac{1}{\max_k \sum_{j=1}^{k} \left| \theta_{k,j}^{[i]} \right|} \right\},$$

provided that $\theta_{k,j}^{[e]} f^{[e]}$ *is replaced with* $\theta_{k,j}^{[e]} \widetilde{f}^{[e]}$ *whenever* $\theta_{k,j}^{[e]}$ *is negative.*

EXAMPLE OF ROS-SSP METHOD

The Ros-SSP3, 2 method derived from Ros-ERK3,2 presented above is given in Table 3.3.1 with its own coefficients.

$\left[\eta_{k,j}\right]$	$\left[\theta_{k,j}^{[i]}\right]$	$\left[\theta_{k,j}^{[e]}\right]$
$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{2}{9}. \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$

TABLE 3.3.1: *Coefficients* $\eta_{k,j}$, $\theta_{k,j}^{[i]}$ *and* $\theta_{k,j}^{[e]}$, $k \geqslant 2, j \geqslant 1$, *of the Ros-SSP3, 2.*

REMARK

One of important applications of the Ros-SSP methods is solving ADR equations. Such ADR equations are often used to simulate physical and biological phenomena. In these problems, the variables are normally positive, e.g., the adsortbate coverage rate of the surface by CO molecules in absorbate-induced phase transition model and the density of bacteria in the chemotaxis model. Hence, as a natural requirement, Ros-SSP methods are expected to preserve the positivity of these variables attaining a demanded order of accuracy.

The diffusion and reaction part in DAR equations can be discretized as in [25] or [3]. The corresponding semi-discretized term which is normally stiff are integrated by choosing a suitable Rosenbrock method such as L-stable ones and therefore no wiggles or spurious oscillation are produced. Consequently, a proper Rosenbrock method does not produce any overshooting values that may deprive the methods of the positivity of numerical results applied to the semi-discretized diffusion and reaction terms .

To guarantee the positivity, the advection part in DAR equations are discretized spatially by a TVD finite difference or discontinuous Galerkin finite element methods. By embedding the SSP method in computing the semi-discretized advection term, the Ros-

SSP methods inherit positivity from the SSP methods [18, 34]. The explicit Runge-Kutta method for the semi-discretized advection term is written as a convex combination of forward Euler steps which are TVD under a suitable Courant-Friedrichs-Levy (CFL) condition. It results characteristic constant which is $\min_{k,j} \frac{\eta_{k,j}}{\left|\theta_{k,j}^{[e]}\right|}$ and the maximum time step-size for positivity of the method is proportional to the above constant.

3.4 Numerical demonstrations

In this section, we test the order of accuracy, the performance, and the robustness of Ros-ERK3,2 and Ros-SSP3,2 via several test problems.

3.4.1 Kaps' problem

The first test problem suggested by Kaps (see in [17]) is given as

$$\frac{dy_1}{dt} = -\left(\varepsilon^{-1} + 2\right) y_1 + \varepsilon^{-1} y_2^2, \tag{3.4.1a}$$

$$\frac{dy_2}{dt} = y_1 - y_2 - y_2^2, \tag{3.4.1b}$$

where $t \in [0, 1]$ and the initial values are $y_1(0) = y_2(0) = 1$. The exact solution for the system is

$$y_1(t) = y_2(t)^2, \qquad y_2(t) = \exp(-t). \tag{3.4.1c}$$

The stiffness of (3.4.1) increases when the positive parameter ε tends to 0. In case of $\varepsilon = 0$, the system degenerates into an index-1 differential algebraic system which is

$$y_1 = y_2^2, \qquad \frac{dy_2}{dt} = y_1 - y_2 - y_2^2.$$

The system is able to be written as of form (3.2.1)

$$\frac{dy}{dt} = f^{[i]}(y) + f^{[e]}(y), \tag{3.4.2a}$$

in which

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad f^{[i]}(y) = \begin{bmatrix} -\varepsilon^{-1} y_1 + \varepsilon^{-1} y_2 \\ 0 \end{bmatrix}, \quad f^{[e]}(y) = \begin{bmatrix} -2y_1 + \varepsilon^{-1} y_2^2 \\ y_1 - y_2 - y_2^2 \end{bmatrix}. \tag{3.4.2b}$$

0	0	0	0
$2-\sqrt{2}$	$(2-\sqrt{2})/2$	$(2-\sqrt{2})/2$	0
1	$\sqrt{2}/4$	$\sqrt{2}/4$	$(2-\sqrt{2})/2$
	$\sqrt{2}/4$	$\sqrt{2}/4$	$(2-\sqrt{2})/2$
	$1/3\,(1-\sqrt{2}/4)$	$\sqrt{2}/4+1/3$	$(2-\sqrt{2})/6$

TABLE 3.4.1: *Butcher table of TR-BDF2 methods*

To solve (3.4.1) we propose two more methods which are TR-BDF2 [5] and Ros-AMF3,2 [25, p. 404]. The former is classical 3-stage, second order L-stable Runge-Kutta method which is a combination between the trapezoidal rule (TR) and a second order backward differentiation formula (BDF2) method. Its coefficients is given in Table 3.4.1. The latter is also a 3-stage, second order L-stable Rosenbrock method in which Jacobian matrices do not require to compute exactly. The computation scheme for Ros-AMF3,2 reads

$$(I-\gamma\Delta tA)\,k_i = \Delta tf\left(y_n+\frac{1}{2}\sum_{j<i}k_j\right)+\Delta tA\sum_{j<i}\gamma_{i,j}k_j,\quad i=1,2,3, \tag{3.4.3a}$$

$$y_{n+1} = y_n+\frac{1}{3}\,(k_1+k_2+k_3)\,, \tag{3.4.3b}$$

where

$$\gamma_{2,1} = -\left(3\gamma+\gamma_{3,1}+\gamma_{3,2}\right),\quad \gamma_{3,2} = \frac{1}{2}-3\gamma,$$

$$\gamma_{3,1} = -\frac{1}{1+2\gamma_{3,2}}\left(6\gamma^3-12\gamma^2+6\left(1+\gamma_{3,2}\right)\gamma+2\gamma_{3,2}^2-\frac{1}{2}\right),$$

$$\gamma = 1-\frac{\sqrt{2}}{2}\cos\theta+\frac{\sqrt{6}}{2}\sin\theta,\quad \theta = \frac{1}{3}\operatorname{atan}\frac{\sqrt{2}}{4},$$

and A is an arbitrary Jacobian approximation of $f'\left(y_n\right)$. All three methods integrate from $t=0$ to 1 using standard automatic step size selection with several given time step size limits. Method TR-BDF2 solves with the original equation (3.4.1) while Ros-AMF3,2 and Ros-ERK3,2 solve the equation (3.4.2).

At first we investigate order of accuracy and stability of Ros-ERK3,2 method by varying ε from 10^{-1} to 10^{-3} and changing time step size from 0.004 to 0.001. Error of Ros-ERK method is estimated by comparing numerical solutions and exact one at $t=1$.

Table 3.4.2 shows us that Ros-ERK3,2 method is of second order of accuracy in all nine test problems. It also means that Ros-ERK3,2 still remains its stability even in very stiff test problems with $\varepsilon = 10^{-3}$.

	$\Delta t = 0.004$	$\Delta t = 0.002$	$\Delta t = 0.001$
$\varepsilon = 10^{-1}$	2.0017	1.9988	2.0008
$\varepsilon = 10^{-2}$	1.9894	1.9917	2.0007
$\varepsilon = 10^{-3}$	2.0568	1.8931	2.0146

TABLE 3.4.2: *Estimated order of accuracy of Ros-ERK3,2 method.*

Then we compare Ros-ERK3,2 with TR-BDF2 and Ros-AMF3,2 in performance aspect. Because the size of test problem (3.4.1) is small, measuring computational time is meaningless. Instead of doing that, we counting the number of computing the right hand side function during the whole integration in the interval $[0, 1]$. The results of three test problems with $\varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ are shown in Table 3.4.3.

	Ros-ERK3,2	Ros-AMF3,2	TR-BDF2
$\varepsilon = 10^{-1}$	1416	1725	407
$\varepsilon = 10^{-2}$	1422	1743	395
$\varepsilon = 10^{-3}$	1482	1821	369

TABLE 3.4.3: *The number of right hand side function calls with error tolerance is* 10^{-6} *and time step size limit is* 10^{-1}.

Table 3.4.3 illustrates the power of TR-BDF2 method in solving extremely stiff problems. Its performance is much more better that these of the others. This fact is easy to understand because Rosenbrock methods are more appropriate to mild and/or very stiff problems rather than the extremely stiff problem in this test. Comparing two methods of Rosenbrock family, Ros-ERK3,2 is better than Ros-AMF3,2. Its number of right hand side function call is roughly 17% less than this number of Ros-AMF3,2.

3.4.2 LINEAR ADVECTION-DIFFUSION PROBLEM

The second test problem is the one-dimensional linear advection-diffusion equation with periodic boundary condition

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = d\frac{\partial^2 u}{\partial x^2}, \quad x \in (0,1), \tag{3.4.4a}$$

$$u(x,0) = \cos(2\pi k x), \quad k \in \mathbb{N}, \tag{3.4.4b}$$

in which a is a velocity and $d > 0$ is a diffusion constant. For the spatial discretization we employ the discontinuous Galerkin methods mentioned in the previous chapter. The semi-discretized system we get is a system of ODEs of the form (3.2.1) in which $f^{[i]}$ and $f^{[e]}$ are from diffusion and advection terms, respectively. As discussed so far, the $f^{[e]}$ is no longer smooth and $f^{[i]}$ is stiff operator. Therefore TR-BDF2 is unable to solve this ODEs even though its performance in solving stiff problems is amazing. We employ Ros-AMF3,2 and the SSP version of Ros-ERK3,2, method Ros-SSP3,2, to to integrate the semi-discretized system of (3.4.4).

The spatial interval $(0,1)$ is divided into small elements whose size is 2^{-7} and on each local element, polynomials up to third degree are used to approximate the exact solution. Therefore, the spatial error is small enough so that it does not largely affect the temporal integration error.

The second order of accuracy of Ros-SSP3, 2 is shown in Table 3.4.4. Here, the parameters in (3.4.4) are chosen as $k = 1$, $a = 1.0$, $d = 0.1$, $t \in (0, 0.5)$.

Time step size	Normalized error norm	Order of accuracy
0.005	$4.3071e-05$	
0.0025	$1.3793e-05$	$1.6428e+00$
0.00125	$3.3945e-06$	$2.0226e+00$
0.000625	$9.3544e-07$	$1.8595e+00$

TABLE 3.4.4: *The errors of Ros-SSP3,2 method at* $t = 0.5$.

Two methods Ros-SSP3,2 and Ros-AMF3,2 are compared with the same time step size limit $\Delta t_{\max} = 0.1$ using the same time step size controller [20]. We take the velocity $a = 1.0$ and test two methods with different values of the diffusion constant $d \in \{1.0, 0.1, 0.01\}$ within the interval $t = [0, 1.0]$. By varying the diffusion constant d from 1.0 to 0.01 and keeping the same velocity $a = 1.0$, the problem (3.4.4) transits

from the diffusion-dominated one to the advection-dominated. The running time of the two methods given in Table 3.4.5 shows that Ros-SSP$_3$, 2 is faster than the Ros-AMF$_3$, 2 during such transition. This outperformance of Ros-SSP$_3$, 2 can be considered as a result of the better stability of the embedded SSP integration method. Such this embedment allows Ros-SSP$_{3,2}$ integrate with bigger time step size and therefore less steps to finish the integration process.

	Elapsed time (sec.)		
d	Ros-AMF$_{3,2}$	Ros-SSP$_{3,2}$	Ratio
1	2.39598	1.42192	40.65%
0.1	2.66305	2.08246	21.80%
0.01	2.81619	2.40671	14.54%

TABLE 3.4.5: *Speed comparison between Ros-SSP$_{3,2}$ and Ros-AMF$_{3,2}$ methods.*

4

Numerical results of practical models

4.1 THE CHEMOTAXIS PROBLEM

Chemotaxis model is a remarkable system of equations in mathematical biology. It is used to mathematically describe [28, 37] the process of pattern formation of Escherichia coli found by Budrene and Berg [7, 8]. As a practical application of Ros-SSP3,2 method, we solve numerically the chemotaxis model presented in [28],

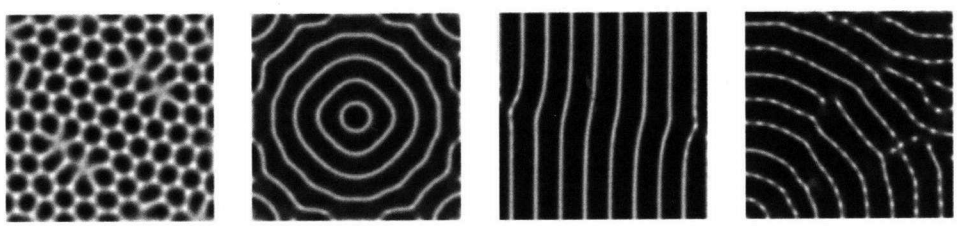
$$\frac{\partial u}{\partial t} = a\Delta u - \mu\nabla\cdot(u\nabla\chi(\rho)) + f(u), \qquad \text{in } \Omega\times(0,\infty), \tag{4.1.1a}$$

$$\frac{\partial \rho}{\partial t} = b\Delta\rho - c\rho + vu, \qquad \text{in } \Omega\times(0,\infty), \tag{4.1.1b}$$

in which the boundary condition is of homogeneous Neumann and initial data is given. Here Ω is a two dimensional bounded domain in which the bacteria are incubated. The unknown functions $u(x, y, t)$ and $\rho(x, y, t)$ denote the population density of the bacteria and the concentration of chemical substance in Ω at time $t \in [0, \infty)$, respectively. The flux of bacteria is described by the term $\mu(u\nabla\chi(\rho))$, where $\chi(\rho)$ denotes a sensitivity function of chemotaxis and $\mu > 0$ denotes a mobility rate of bacteria. $a > 0$ and $b > 0$

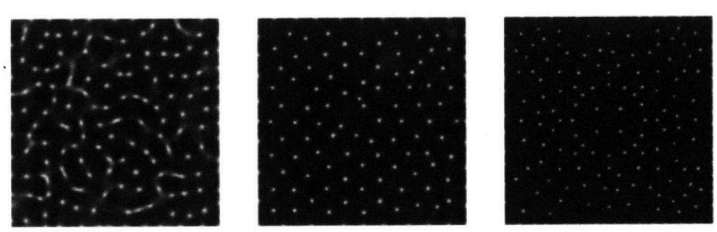
are the diffusion rates of bacteria and chemical substance, respectively. $c > 0$ and $v > 0$ are the degradation and production rates of ρ, respectively. The growth of the bacteria is determined via $f(u)$.

All parameters for numerical computation are taken as same as in [40, p. 353]. As discussion in previous examples, the robustness of Ros-SSP3,2 method is proven via re-creating a series of patterns by varying the chemotaxis coefficient μ from small to relatively large. This series of patterns contains honeycomb, swarm rings, continuous stripes, and perforated stripes for small values of μ as in Figure 4.1.1. When μ becomes larges the series is augmented with chaotically moving short-perforated-lines, chaotically moving dots, and stabilized isolated dots patterns shown in Figure 4.1.2.



(A) *Honeycomd* (B) *Swarm rings* (C) *Continuous stripes* (D) *Perforated stripes*

FIGURE 4.1.1: *Values of μ for these patterns are 6.0, 7.2, 8.2, and 8.5, respectively.*



(A) *Chaotically moving perforated lines* (B) *Chaotically moving dots* (C) *Stabilized dots*

FIGURE 4.1.2: *Values of μ for these patterns are 10, 15, and 40, respectively.*

4.2 TUMOR-INDUCED ANGIOGENESIS MODEL

The tumor-induced angiogenesis model introduced by Anderson and Chaplain [1] describes the formation of blood vessel from a pre-vasculature during the growth of solid tumors. This model consists of three variables which are the density of endothelial cells, the concentration of tumor angiogenic factors (TAF), and the concentration of fibronectin. Initially, TAFs secreted by a solid tumor diffuse and create a chemical gradient to neighboring blood vessels. The endothelial cells lining these vessels response chemotactically to these factors and begin to migrate towards the tumor. In the migration, the cells have to pass through the extracellular matrix containing fibronectin. Fibronectin is also synthesized and secreted by the endothelial cells and it stimulates the directed migration of the endothelial cells. This response of the cells to the gradient of fibronectin is termed haptotaxis. Therefore, the movement of the endothelial cells is affected by two elements. One is the chemotactic effect caused by TAF produced from the solid tumor and the other is the haptotactic one caused by fibronectin on the extracellular matrix and from the cells themselves. These processes are mathematically modeled by a diffusion-advection equation in a two-dimensional bounded domain Ω,

$$\frac{\partial n}{\partial t} = D_n \Delta n - \mu \nabla \cdot (n \nabla \chi(c)) - \nu \nabla \cdot (n \nabla \psi(f)) + \delta n^2 \left(1 - \frac{n}{\kappa}\right), \tag{4.2.1a}$$

$$\frac{\partial f}{\partial t} = \beta n - \gamma n f, \tag{4.2.1b}$$

$$\frac{\partial c}{\partial t} = -\zeta c - \eta n c, \tag{4.2.1c}$$

in which the boundary condition is of homogeneous Neumann and initial data is given. The unknown function $n(x, y, t), f(x, y, t)$, and $c(x, y, t)$ are the density of endothelial cells, the concentration of fibronectin, and the concentration of TAF, respectively. This model is modified from the continuous one proposed in [1].

The evolution equation of the endothelial cells, (4.2.1a), consists of four terms in its right hand side. The first term is the natural diffusion of cells with a diffusion parameter $D_n > 0$. The chemotactic migration of cell is described by the term $\mu \nabla \cdot (n \nabla \chi(c))$, where $\mu > 0$ is a chemotaxis parameter and $\chi(c)$ is a chemotaxis function. The haptotaxis behavior is expressed via the term $\nu \nabla \cdot (n \nabla \psi(f))$, where $\nu > 0$ is a haptotaxis parameter and $\psi(f)$ is a haptotaxis function. The last term in the right hand side of (4.2.1a) is the proliferation of the endothelial cells in which $\kappa > 0$ is a capacity for

the cells and $\delta > 0$ is a growth rate of cells. The second equation (4.2.1b) describes the growth of fibronectin in which $\beta > 0$ is a production rate of fibronectin produced by the endothelial cells and $\gamma > 0$ is an uptake parameter illustrate the uptake and binding of fibronectin to the endothelial cells as they grow towards to the solid tumor. The last equation (4.2.1c) describes the evolution of TAF consisting of the natural decay with a rate $\zeta > 0$ and some uptake by cells by a rate $\eta > 0$.

The method Ros-SSP3,2 is used to solved the model equation (4.2.1) with all fixed parameters

$$D_n = 0.00035,\ \mu = 0.4,\ \delta = 0.1,\ \kappa = 1.0,\ \beta = 0.05,\ \gamma = 0.1,\ \zeta = 0.05,\ \eta = 0.1,$$

except the haptotaxis one, v. The chemotaxis and haptotaxis functions are both chosen as linear ones,

$$\chi(c) = c \text{ and } \psi(f) = f.$$

The initial profile is given in the domain $\Omega = [0,1] \times [0,1]$ as

$$n(x,y,0) = \begin{cases} \exp\left(-\frac{x^2}{0.001}\right)\cos(6\pi y) & \text{if } \frac{2k+1}{12} \leqslant y \leqslant \frac{2k+3}{12},\ k = 0,2,4, \\ 0 & \text{otherwise,} \end{cases}$$

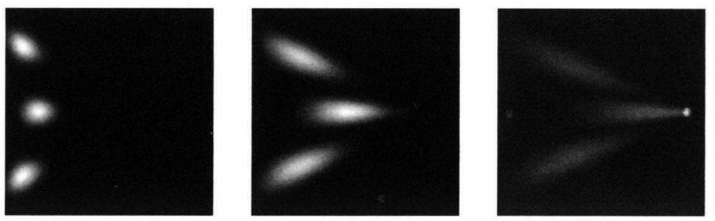
$$f(x,y,0) = 0.75\exp\left(-\frac{x^2}{0.45}\right),$$

$$c(x,y,0) = \begin{cases} 1 & \text{if } r \leqslant 0.1, \\ (1.1-r)^2 & \text{otherwise,} \end{cases}$$

with $r = \sqrt{(x-1)^2 + (y-0.5)^2}$. The spatial discretization is the same as for the above example with element size 2^{-5} and local approximation polynomials are of second degree. The RosSSP3,2 is employed for temporal integration.

At first, we consider the case of no haptotaxis, i.e., $v = 0$. Under the effect of chemotaxis phenomena only, the endothelial cells migrates quickly and directly through the extracellular matrix to the solid tumor by $t = 2.5$ as shown in Figure 4.2.1.

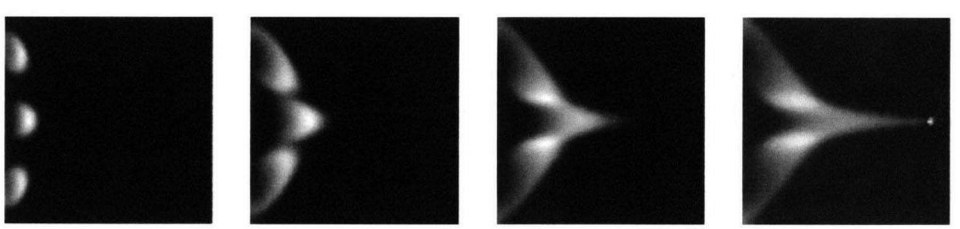
If the haptotaxis effect is included by setting $v = 0.25$, the movement of cells changes as in Figure 4.2.2. It can be seen that the migration of cells toward the solid tumor is slower than the case without haptotaxis. The aggregation of cells under the effect of



(A) $t = 1.0$ (B) $t = 2.0$ (C) $t = 2.5$

FIGURE 4.2.1: *Numerical simulation of the evolution of the endothelial cell density without haptotaxis effect. The color graduation is proportional to the cell density, white is high density and black low density.*

fibronectin can be observed clearly as the forming clusters at $t = 3.0$. After that, these clusters merge together to form one cluster with high density of endothelial cells ($t = 4.5$). And this cluster migrates slowly to the solid tumor at $t = 5.5$.



(A) $t = 1.0$ (B) $t = 3.0$ (C) $t = 4.5$ (D) $t = 5.5$

FIGURE 4.2.2: *Numerical simulation of the evolution of the endothelial cell density with haptotaxis effect. The color graduation is proportional to the cell density, white is high density and black low density.*

References

[1] A. Anderson and M. Chaplain, *Continuous and discrete mathematical models of tumor-induced angiogenesis*, Bulletin of Mathematical Biology, 60 (1998), pp. 857–899. 10.1006/bulm.1998.0042.

[2] Douglas N. Arnold, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19 (1982), pp. 742–760.

[3] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2001/02), pp. 1749–1779.

[4] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Appl. Numer. Math., 25 (1997), pp. 151–167. Special issue on time integration (Amsterdam, 1996).

[5] R.E. Bank, W.M. Coughran, W. Fichtner, E.H. Grosse, D.J. Rose, and R.K. Smith, *Transient simulation of silicon devices and circuits*, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 4 (1985), pp. 436 – 451.

[6] F. Bassi and S. Rebay, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys., 131 (1997), pp. 267–279.

[7] Elena O. Budrene and Howard C. Berg, *Complex patterns formed by motile cells of escherichia coli*, Nature, 349 (1991), pp. 630–633.

[8] ——, *Dynamics of formation of symmetrical patterns by chemotactic bacteria*, Nature, 376 (1995), pp. 49–53.

[9] Guy Chavent and Bernardo Cockburn, *The local projection P^0P^1-discontinuous-Galerkin finite element method for scalar conservation laws*, RAIRO Modél. Math. Anal. Numér., 23 (1989), pp. 565–592.

[10] G. Chavent and G. Salzano, *A finite-element method for the 1-D water flooding problem with gravity*, J. Comput. Phys., 45 (1982), pp. 307–344.

[11] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case*, Math. Comp., 54 (1990), pp. 545–581.

[12] Bernardo Cockburn, San Yih Lin, and Chi-Wang Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems*, J. Comput. Phys., 84 (1989), pp. 90–113.

[13] Bernardo Cockburn and Chi-Wang Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework*, Math. Comp., 52 (1989), pp. 411–435.

[14] ——, *The Runge-Kutta local projection P^1-discontinuous-Galerkin finite element method for scalar conservation laws*, RAIRO Modél. Math. Anal. Numér., 25 (1991), pp. 337–361.

[15] ——, *The local discontinuous Galerkin method for time-dependent convection-diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440–2463 (electronic).

[16] ——, *The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems*, J. Comput. Phys., 141 (1998), pp. 199–224.

[17] K. Dekker and J. G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, vol. 2 of CWI Monographs, North-Holland Publishing Co., Amsterdam, 1984.

[18] Sigal Gottlieb and Chi-Wang Shu, *Total variation diminishing Runge-Kutta schemes*, Math. Comp., 67 (1998), pp. 73–85.

[19] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Rev., 43 (2001), pp. 89–112 (electronic).

[20] Kjell Gustafsson and Gustaf Söderlind, *Control strategies for the iterative solution of nonlinear equations in ODE solvers*, SIAM J. Sci. Comput., 18 (1997), pp. 23–40. Dedicated to C. William Gear on the occasion of his 60th birthday.

[21] E. Hairer and G. Wanner, *Solving ordinary differential equations. II*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 1996. Stiff and differential-algebraic problems.

[22] Jan S. Hesthaven and Tim Warburton, *Nodal discontinuous Galerkin methods*, vol. 54 of Texts in Applied Mathematics, Springer, New York, 2008. Algorithms, analysis, and applications.

[23] M. Hildebrand, M. Kuperman, H. Wio, A. S. Mikhailov, and G. Ertl, *Self-organized chemical nanoscale microreactors,* Phys. Rev. Lett., 83 (1999), pp. 1475–1478.

[24] W. Hundsdorfer, B. Koren, M. van Loon, and J. G. Verwer, *A positive finite-difference advection scheme,* J. Comput. Phys., 117 (1995), pp. 35–46.

[25] Willem Hundsdorfer and Jan Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations,* vol. 33 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2003.

[26] Christopher A. Kennedy and Mark H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations,* Appl. Numer. Math., 44 (2003), pp. 139–181.

[27] Randall J. LeVeque, *Finite volume methods for hyperbolic problems,* Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2002.

[28] Masayasu Mimura and Tohru Tsujikawa, *Aggregating pattern dynamics in a chemotaxis model including growth,* Physica A: Statistical Mechanics and its Applications, 230 (1996), pp. 499 – 543.

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

[30] P.-O. Persson and J. Peraire, *Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations,* SIAM J. Sci. Comput., 30 (2008), pp. 2709–2733.

[31] Ronald F. Probstein, *Physicochemical Hydrodynamics: An Introduction,* A Wiley–Interscience Publication, 1994.

[32] H. H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations,* Comput. J., 5 (1962/1963), pp. 329–330.

[33] Khosro Shahbazi, *An explicit expression for the penalty parameter of the interior penalty method,* Journal of Computational Physics, 205 (2005), pp. 401 – 407.

[34] Chi-Wang Shu and Stanley Osher, *Efficient implementation of essentially nonoscillatory shock-capturing schemes,* J. Comput. Phys., 77 (1988), pp. 439–471.

[35] J. W. Thomas, *Numerical partial differential equations: finite difference methods,* vol. 22 of Texts in Applied Mathematics, Springer-Verlag, New York, 1995.

[36] VIDAR THOMÉE, *Galerkin finite element methods for parabolic problems*, vol. 25 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, second ed., 2006.

[37] REBECCA TYSON, *Pattern formation by E. coli mathematical and numerical investigation of a biological phenomenon*, PhD thesis, University of Washington, Seattle, WA, 1996.

[38] REBECCA TYSON, L. G. STERN, AND RANDALL J. LEVEQUE, *Fractional step methods applied to a chemotaxis model*, J. Math. Biol., 41 (2000), pp. 455–475.

[39] J. G. VERWER, E. J. SPEE, J. G. BLOM, AND W. HUNDSDORFER, *A second-order Rosenbrock method applied to photochemical dispersion problems*, SIAM J. Sci. Comput., 20 (1999), pp. 1456–1480.

[40] ATSUSHI YAGI, *Abstract parabolic evolution equations and their applications*, Springer Monographs in Mathematics, Springer-Verlag, Berlin, 2010.

List of publications

1. Doan Duy Hai and Atsushi Yagi, *Numerical computations and pattern formation for chemotaxis-growth model,* Sci. Math. Jpn., **70**, 2009, pp. 205-211.

2. Doan Duy Hai and Atsushi Yagi, *Longtime behavior of solutions to chemotaxis-proliferation model with three variables,* AIMS Disc. and Cont. Dynam. Syst., **32**, 2012, pp. 3957-3974.

3. Doan Duy Hai and Atsushi Yagi, *Rosenbrock strong stability-preserving methods for convection-diffusion-reaction equations,* Japan J. Indust. Appl. Math. sumitted.

4. Doan Duy Hai and Atsushi Yagi, *Longtime behavior of solutions to some mathematical model for angiogenesis,* in preparation.