Block Splitting of Exponential Operators via Padé Approximation and BSPTS Schemes

by

Bin Yin, B.Math.

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Science

School of Mathematics and Statistics

Ottawa-Carleton Institute for Mathematics and Statistics

Carleton University

Ottawa, Ontario, Canada

May 2006

Your file Votre référence
ISBN: 978-0-494-16507-2
Our file Notre référence
ISBN: 978-0-494-16507-2

Block Splitting of Exponential Operators via Padé Approximants and BSPTS Schemes

Bin Yin

School of Mathematics and Statistics,

Carleton University, Ottawa, Ontario, K1S 5B6

Abstract

A new class of explicit schemes, Padé time-stepping (PTS) schemes has been recently created and introduced [4] for numerically evolving solutions of ordinary differential equations problems. PTS schemes, which are as simple and inexpensive per time-step as other explicit methods, possess, in many cases, properties of stability similar to those offered by implicit approaches. However, as noted in [4], PTS schemes are not unconditionally stable in general, and in certain cases they are even less stable than some other popular explicit finite difference schemes.

In this thesis we look at the role of a generalization of rational PTS techniques, which aims to provide increased stability over a wider range of problems, while at only a limited additional cost. The Block Splitting Padé Time-stepping schemes (BSPTS) we introduce in this context, are based on the operator splitting method combined with Padé approximation. To increase the efficiency and stability of the numerical solutions, the operator splitting method is employed to partition the operator into M additive blocks. Each block usually is multicomponent, instead of single-component in PTS. By doing such splitting, there are at least

two advantages that can be gained.

Firstly, the size of block can be chosen to satisfy a diverse range of approach goals, in terms of efficiency, stability and accuracy. BSPTS can achieve unconditional stability in certain cases where PTS schemes are not. While requiring extra computational cost on a per-step basis, BSPTS schemes can still provide overall gains in efficiency relative to PTS.

The other advantage is that, the BSPTS schemes improve higher order block splitting methods applicable to certain problems, for example parabolic PDEs. As it is well known [26], block splitting methods are unstable in higher order (order greater than 2) for parabolic PDEs. For higher order BSPTS schemes in evolving certain problems, an appropriate splitting factor s is required and which can be chosen, in such a way that eigenvalues of each block are negative, therefore the 'partial' solution by solving each block is convergent. Consequently, the solution obtained from overall higher order splitting methods is convergent and stable.

Acknowledgments

I would like to thank my supervisor, Dr. David E. Amundsen, who has been helping me when I am in trouble, encouraging me when I lose confidence and sharing happiness with me when I make progress. This thesis would not come out without the invaluable suggestion and patient guidance from Dr. Amundsen. If not for him, I would not have learned so much. My thanks also go out to the School of Mathematics and Statistics, Carleton University. Thanks to all staff and friends who support me during these years.

I thank my family for all the support they gave me over the previous years and this year in particular.

Contents

List of Figures

Chapter 1

Introduction

The numerical methods for solving time-dependent ordinary differential equations (ODE) and partial differential equations (PDE), in large part, can be divided into explicit schemes and implicit schemes. Explicit schemes tend to be the most efficient methods per step for solving these types of problems, but the price we have to pay is the stiffness and restriction on time-step size for stability. Implicit schemes have an advantage on this since there tends to be far less and in some cases no restriction on step size chosen to reach stability. However, implicit schemes are typically much more expensive on a per-step basis which again leads to significant computational cost.

This has motivated the development of strongly-stable explicit methods, or on the other hand, efficient implicit schemes. There are numerous examples of such approaches, for instance, the classical implicit Runge-Kutta methods [12]. However, with accuracy higher than second order they require recursive multiple evaluations within a time step and are often therefore cumbersome to implement. A significant amount of additional storage is required and the resulting

linear systems tend to be larger. Similar problems arise in methods which utilize higher gradients. In the specific context of linear cases, Argyris et al. [5] introduced a method involving higher time derivatives. However, in applications to nonlinear problems, higher time derivatives require repeated applications of the chain rule leading to higher space derivatives and therefore to lengthy expressions which are expensive to compute and cumbersome to linearize. For predictor-corrector methods [24], an explicit scheme is used to produce an initial guess for the implicit solve; however, for stability full nonlinear solution is required in the corrector step which in general leads to high computational costs. Explicit Strong-Stability-Preserving Runge-Kutta (SSPRK) methods [27] are a specific type of time integration method that are widely used to integrate hyperbolic PDEs. Under a suitable step-size restriction, these methods share a desirable nonlinear stability property with the underlying PDE; e.g., stability with respect to total variation or maximum norm. But there is a barrier that the SSPRK with positive coefficients fundamentally restrict the achievable CFL coefficient for linear, constant-coefficient problems and the overall order of accuracy for general nonlinear problems.

Other methods include split-step [3], semi-implicit [9] schemes, as well as the exponential proagation method [16]. Once again, all those examples have their advantages in some cases, but also, have their shortcomings in certain aspects. For example, split-step and semi-implicit methods can be attractive under certain circumstances, since they can derive stability from the use of exact or otherwise easily obtainable solutions for some portions of the differential operator at hand; of course, the existence of such useful easily-obtainable solutions cannot be guaranteed in general [4]. The exponential proagation method evolves the solution through linearization and evaluation of matrix exponentials. In turn, in

cases of large matrices, this method may lead to unpractical CPU and memory costs.

It is interesting to note that all of these schemes bear connections with Padé approximation [6]. For instance, the classical implicit methods can be expressed in terms of Padé approximants with matrix valued arguments [15]; the matrix exponential required by the exponential propagation method, on the other hand, are frequently obtained via Padé approximation of matrix valued Taylor series [16].

A new class of explicit schemes, Padé time-stepping schemes (PTS) has been recently created and introduced [4]. PTS schemes, which are as simple and inexpensive per time-step as other explicit methods, possess, in many cases, properties of stability similar to those offered by implicit approaches. However, as noted in [4], PTS schemes are not unconditionally stable in general, and in certain cases, they are even less stable than some other popular explicit schemes, such as, Runge-Kutta methods.

In this thesis we look at the role of a generalization of rational PTS techniques, which aims to provide increased stability over a wider class of problems, while at only a limited additional cost. The Block Splitting Padé Time-stepping schemes (BSPTS) we introduce in this context, are based on the operator splitting method combined with Padé approximation. To increase the efficiency and stability of the numerical solutions, the operator splitting method is employed to partition the operator into M additive blocks. A Padé based stepping scheme is then applied to each block.

By doing such splitting, we find that at least two advantages can be gained.

Firstly, the size of block can be chosen to satisfy a diverse range of approach goals, in terms of efficiency, stability and accuracy. Depending on the structure of the operators, an optimal M is associated with the least computational time per step while the overall scheme is stable. Obviously, the whole operator can be treated as a single block, which is what we will call Matrix PTS (MPTS) schemes, a class of strongly stable schemes. In turn, if M is chosen to be the dimension of the entire operator, then we obtain the componentwise PTS schemes, which can achieve significant efficiency on a per-step basis. An optimal BSPTS scheme, presented as an 'intermediate' approach between componentwise PTS and implicit MPTS, will not only inherit the consistency of PTS schemes, but benefit from association with the strong stability of implicit schemes. BSPTS can achieve unconditional stability in certain cases where PTS schemes are not. While requiring limited extra computational cost on a per-step basis, BSPTS schemes can still provide overall gains in efficiency relative to PTS.

The other achievement is that, the BSPTS schemes improve higher order block splitting methods applicable to certain problems, for example parabolic PDEs. As is well known [26], block splitting methods are unstable in higher order (order greater than 2) for certain cases, especially parabolic PDEs. In our study, the consistency of Padé approximation is added into each block and overall splitting scheme. For higher order BSPTS schemes, an appropriate splitting factor s is required and which can be chosen, in such a way that eigenvalues of each block are negative, therefore the 'partial' solution by solving each block is convergent. Consequently, the solution obtained from overall higher order splitting methods is convergent and stable.

This thesis is organized as follows. In Chapter 2, we briefly introduce the Padé

approximation techniques and PTS, as well the implementation issues on PTS schemes. In Chapter 3, we extend the introduction to the Padé approximants of matrices and MPTS schemes; we also state a fundamental theorem on the unconditional stability of MPTS. In Chapter 4, the idea of general exponential operator splitting is presented, and we state some well-known results in the field, including the efficiency, accuracy, convergency, etc. In Chapter 5, an idea of exponential operator splitting via Padé approximation, and consequently, the BSPTS schemes are introduced more precisely. The stability analysis shows that the BSPTS method is unconditionally stable in a wide range of linear cases, such as diagonal operators and multi-diagonal operators. Various numerical experiments on linear problems illustrate the performance of BSPTS methods, compared to other techniques. In particular the schemes were tested on a well known stiff problem, the classical KdV equation, and it is seen BSPTS gives an improvement in the stability over PTS schemes and other explicit methods.

Chapter 2

Padé Time-Stepping Schemes

Padé time-stepping (PTS) scheme is a new class of explicit schemes based on use of one-dimensional Padé approximation [6]. It exhibits the desirable simplicity and reduced operation count of other explicit schemes, and it also possesses the property of unconditional stability in some cases. The following sections provide a brief overview of the method and its key implemention issues.

2.1 Padé time-stepping schemes (PTS)

Time-stepping numerical schemes for a system of differential equations

$$\dot{\vec{u}} = A\vec{u}, \quad \vec{u}(t=0) = \vec{u_0}, \tag{2.1}$$

are generally based on Taylor expansions of the unknowns and correspond to using certain approximation arguments. The PTS approach, time-steps the solution by means of Padé approximations for each one of the scalar unknowns. As is well known, the $[L/M]$-Padé approximant of a function $u = u(t)$ is defined [6]

as a rational function of polynomials in t with numerator degree L, denominator degree M, and whose Taylor series agrees with that of u up to order $L+M$ [6].

Assuming the solution u of (2.1) is sufficiently differentiable, the Taylor series of the solution, in time-step h at fixed t, may be written as

$$u(t+h) = \sum_q c_q(t) h^q. \tag{2.2}$$

A variety of approaches can be applied to produce such series numerically; here are two examples of such algorithms [4]. The simplest and most direct way to evaluate Taylor expansions from a given differential equation (2.1) with a given value $u(t_0)$ proceeds via an inductive calculation similar to that arising in the proof of the Cauchy-Kowalevsky theorem [19]. By this inductive procedure, the coefficients of the Taylor expansions (2.2) can be generated via multiple differentiation of the equations with respect to t and spatial variables x, if any. The resulting Taylor series by this method is adequate for problems in which the derivatives of the equation are easy to obtain. For example, for the convection-diffusion equation

$$u_t + \alpha u_x = \nu u_{xx}, \quad \alpha \in R, \quad \nu \in R^+ \tag{2.3}$$

Suppose the value of u at $t = t_0$ is known to be

$$u(x, t_0) = f(x) \text{ for } x \in [a, b]. \tag{2.4}$$

Then, (2.3) and (2.4) give

$$u_t(x, t) = -\alpha f'(x) - \nu f''(x) \text{ for } x \in [a, b]. \tag{2.5}$$

And if the second time-derivatives of u is required, we can differentiate the equation (2.3) to obtain

$$u_{tt} = -\alpha u_{xt} + \nu u_{xxt}, \tag{2.6}$$

where all terms of the right hand side of (2.6) can be obtained from (2.4). Clearly the Taylor series of form (2.2) for this problem can be produced by indefinitely continuing this procedure.

On the other hand, for certain types of applications, it may be highly preferable to use the 'multi-step' PTS algorithms where the Taylor coefficients are evaluated on the basis of values of the equation, or say the local value of its operator only. More details about multi-step PTS applied to the numerical differentiation are discussed in [4].

The PTS scheme evolves the solution u by means of the Padé approximant of the Taylor series (2.2) for each component m of the vector valued unknown function. By calling h the time-step, for a given pair of integers (L, M) we denote the Padé approximant to the Taylor series (2.2) by

$$[L/M]^{m,t}(h) = \frac{a_0^{m,t} + a_1^{m,t}h + \cdots, +a_L^{m,t}h^L}{1 + b_1^{m,t}h + \cdots, +b_M^{m,t}h^M} = \frac{P_L(h)}{Q_M(h)}. \tag{2.7}$$

In general, the coefficients a_n, b_n are determined from the Taylor series expansion of $u(t+h)$ at any regular point. If, without loss of generality, the expansion is taken as $t = 0$, so

$$[L/M]^{m,t}(h) = \sum_{n=0}^{L+M} c_n h^n + O(h^{L+M+1}), \tag{2.8}$$

where c_n are known. From the algebraic system

$$\frac{P_L(h)}{Q_M(h)} + O(h^{L+M+1}) = \sum_{n=0}^{L+M} c_n^m h^n, \tag{2.9}$$

the coefficients of $h^{L+1}, h^{L+2}, \ldots, h^{L+M}$ directly yield the coefficients

$$\begin{pmatrix} b_M \\ b_{M-1} \\ \vdots \\ b_1 \end{pmatrix} = - \begin{pmatrix} c_{L-M+1} & c_{L-M+2} & \cdots & c_L \\ c_{L-M+2} & c_{L-M+3} & \cdots & c_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ c_L & c_{L+1} & \cdots & c_{L+M+1} \end{pmatrix}^{-1} \begin{pmatrix} c_{L+1} \\ c_{L+2} \\ \vdots \\ c_{L+M} \end{pmatrix}, \tag{2.10}$$

where $c_n \equiv 0$ for $n < 0$. If the inverse matrix of c_n elements does not exist then the particular $[L/M]$ Padé approximatant (2.7) is not defined [2, 6]. Once the b_n are found, then the remaining equations $(h^0, h^1, \ldots, h^L)$ yield the a_n coefficients.

Provided (2.7) exists, then, calling $t_j = jh$, the j-th time-step ($j = 1, 2, \cdots$), the $[L/M]$ Padé time-stepping scheme (PTS$[L/M]$) time-steps the solution according to the prescription

$$u_{j+1}^m = [L/M]^{m,t_j}(h) \tag{2.11}$$

unless one of the following situations occur:

1. The Padé approximant $[L/M]^{m,t_j}(h)$ does not exist, or

2. The Padé approximant $[L/M]^{m,t_j}(h)$ exists but the condition

$$|[L/M]^{m,t_j}(h)| < K|u_j^m| \tag{2.12}$$

is violated - where K is an appropriately large constant, and problem dependent. This situation may occur when, in some cases, the Padé denominator vanishes or is very small, In other words, $|Q_M(h)| < \epsilon$, where ϵ is a small constant.

Thus the condition 2 above is equivalent to

2′. The Padé approximant $[L/M]^{m,t_j}(h)$ exists but, the condition

$$|Q_M(h)| > \epsilon \tag{2.13}$$

is violated.

Note that, even under certain conditions, the PTS scheme may still become 'locally inaccurate', that is, it may produce, at certain time-steps, accuracies of

an order lower than that of the underlying Taylor series. Since to discuss the 'locally inaccurate' of PTS schemes is not the focus of thesis, please refer to [4] for details.

2.2 Stability of Padé time stepping schemes

The stability properties of a time-stepping scheme are usually drawn from consideration of its A-stability, that is, the stability properties of the scheme when applied to a linear system. However, given the nonlinear nature of Padé approximation it is impossible to deduce a property of unconditional stability for a general linear system of ODEs from corresponding properties of stability for single linear ODEs. In fact, PTS schemes are not unconditionally A-stable.

Nevertheless some general stability properties of PTS schemes can be obtained through application to linear systems. While in general for a given order of accuracy several PTS$[L/M]$ schemes could be considered for different values of L and M, this fact along with numerical experiments show that balanced schemes, where $L \approx M$, tend to provide more favorable stability properties. Henceforth we restrict our focus on PTS schemes with $L = M$ which are called diagonal PTS schemes. In particular, for single linear equations, diagonal PTS schemes are always unconditionally stable.

It has been well known that time stepping a single linear equation by means of Padé approximants gives rise to unconditionally A-stable numerics [8], and Amundsen and Bruno [4] have extended this result to triangular linear systems with negative diagonal entries. Then in our nomenclature, we state

Theorem 1. *Let A be a real, upper triangular $k \times k$ matrix with negative diagonal entries. Then, for every value of N, the PTS[N/N] scheme for the system*

$$\dot{u} = Au,$$

is stable for all stepsizes $h > 0$. More precisely, for all $h > 0$ we have

$$u_j^m \to 0 \text{ as } j \to \infty \text{ for } 1 \leq m \leq k.$$

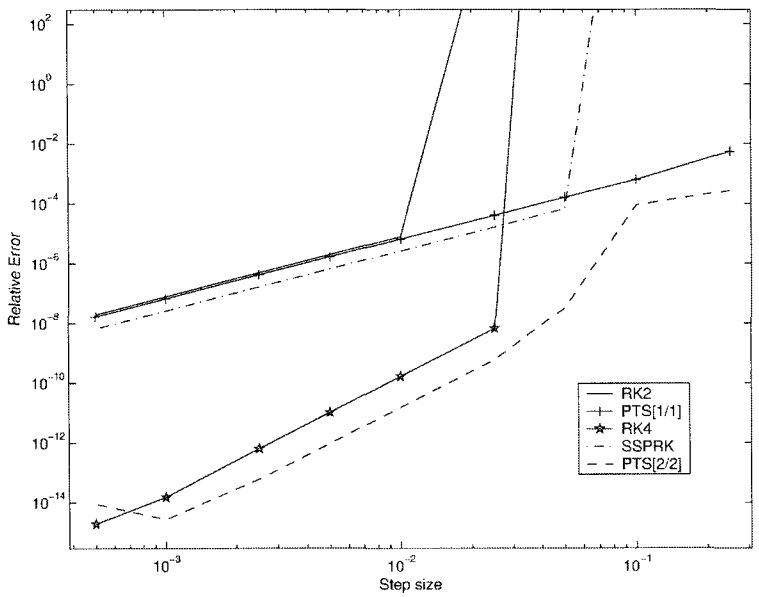

Figure 2.1: Accuracies of PTS[1/1] and PTS[2/2] schemes applied to a system of ordinary differential equations with an uppertriangular operator, comparing to that of RK2, RK4 and SSPRK methods

However, for the general systems of differential equations, it has been concluded [4] that PTS schemes, in general, are not unconditionally stable and depended on the nature of the matrix itself.

To illustrate the behaviour of PTS schemes, we present some simple examples. In our first numerical example, the PTS[1/1] and PTS[2/2] are applied to an upper triangular matrix associated with a 4×4 system of ordinary differential

equations

$$\vec{u'} = \begin{bmatrix} -100 & 1 & -1 & 1 \\ 0 & -10 & 1 & 1 \\ 0 & 0 & -2 & 1 \\ 0 & 0 & & -0.1 \end{bmatrix} \vec{u} \quad ; \quad \vec{u}(0) = \begin{bmatrix} 0.01 \\ 0.1 \\ 1 \\ 10 \end{bmatrix}. \tag{2.14}$$

In this example, clearly the disparate eigenvalues make this problem ill-conditioned for implementation of regular explicit methods. As we can see in Figure 2.1, the classical RK2, RK4, and even the strongly stable SSPRK scheme all lose their stability at certain stepsizes as the associated relative errors[1] are greater than 1. On the other hand, PTS[1/1] and PTS[2/2] not only perform as well as RK schemes at small step-sizes, but also provide accurate results while the solutions given by RK schemes are blowing up, which confirms the Theorem 1.

In what follows we present two numerical experiments, which may demonstrate the 'two-side' stability properties of PTS schemes. Firstly, we consider a tri-diagonal linear system

$$\vec{u'} = \begin{bmatrix} -1000 & -1 & 0 \\ 1 & -10 & -1 \\ 0 & 1 & -5 \end{bmatrix} \vec{u} \quad ; \quad \vec{u}(0) = \begin{bmatrix} 0.1 \\ 1 \\ 100 \end{bmatrix}. \tag{2.15}$$

The results are shown in Figure 2.2. For this tridiagonal operator with disparate eigenvaues, PTS[1/1] and PTS[2/2] schemes also provide accurate results at relatively large step-sizes, where RK schemes are unstale. In the next experiment we apply the PTS schemes to a large system of ODEs arising from discretization

[1] Throughtout this thesis, a *relative error* is defined as the ratio of the norm of the difference between the numerical solution u_n and the exact solution u_e to the norm of the exact solution, i.e. $\frac{|u_n - u_e|}{|u_e|}$

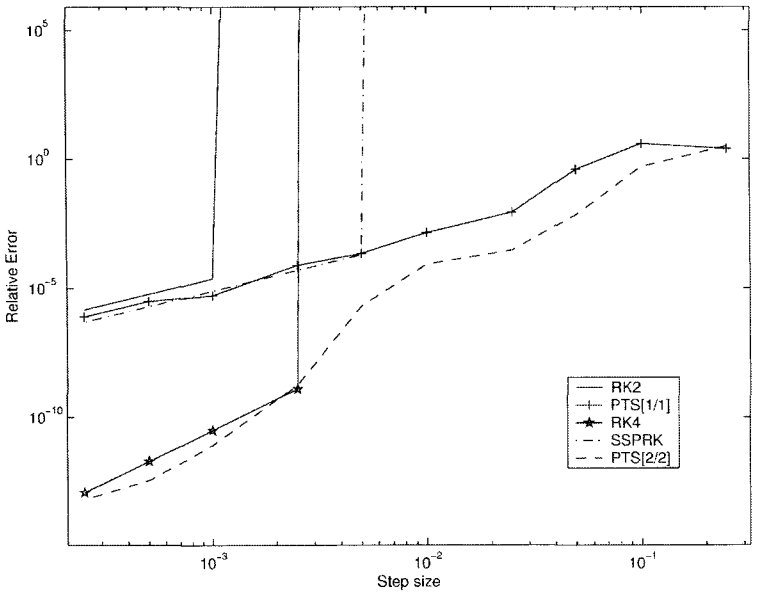

Figure 2.2: Stability of PTS schemes vs. various explicit methods on a system of ODEs with a tridiagonal operator

of time-dependent hyperbolic PDE. From this example, it can be seen that the properties of the PTS schemes in PDE applications also depend significantly on the type of spatial discretizations used.

Next we consider the one way wave equation

$$u_t + u_x = 0 \tag{2.16}$$

with periodic boundary conditions and initial condition $u(x,0) = e^{-x^2}$ on the domain $x \in [-5,5]$. The corresponding system of ODEs is taken as

$$u_t = Au$$

where A is the circulant matrix corresponding to the centered difference representation for u_x, such that

$$u_x \approx \frac{u_{j+1} - u_{j-1}}{2\Delta x}, \tag{2.17}$$

where u_{j+1} and u_{j-1} are results at $(j+1)^{st}$ and $(j-1)^{st}$ spatial points, respec-

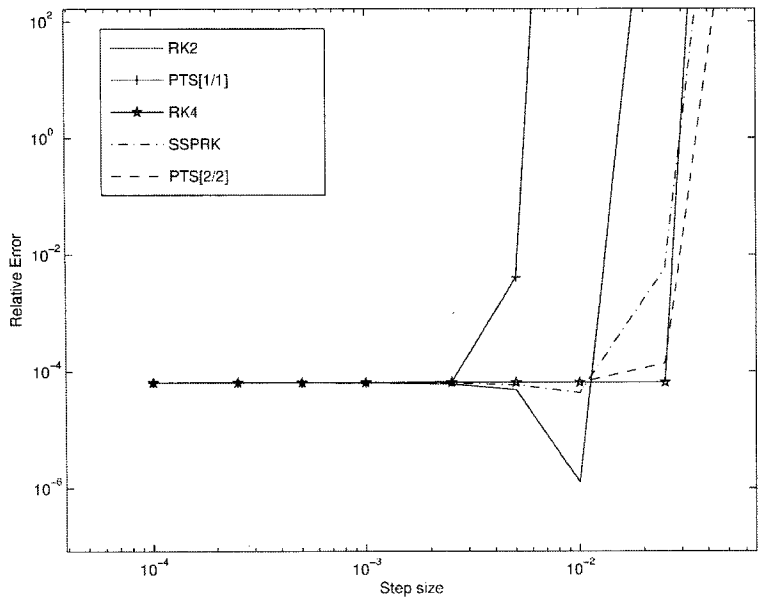

Figure 2.3: Convergence for various explicit methods on one way wave equation $u_t + u_x = 0$, with initial condition $u(x,0) = e^{-x^2}$

tively; a spatial mesh-spacing $\Delta x = 0.01$ is used in this case.

Although the matrix A is of similar nature to the one in the previous example, but just of larger size, as we can see in Figure 2.3, in this case PTS[1/1] and PTS[2/2] schemes are not only unstable for relatively large step sizes, but even less stable than RK schemes. In fact, in this case, PTS[1/1] is less stable than RK2.

From context above and [4] we know that, for a wide range of evolution problems, the use of the PTS schemes can give rise to higher efficiency compared to that offered by leading time-stepping methods. However, in a number of important cases PTS does not exhibit such desirable properties, e.g. the example above. In those cases, in order to procure stability but at greater computational cost, we can apply a matrix reformulation as will be discussed in the next chapter.

Chapter 3

Matrix PTS Schemes

In the previous chapter, we have discussed the PTS schemes with component-wise computation, which can achieve desirable efficiency on a per time-step basis and possess, in many case, properties of stability similar to those offered by implicit approaches [4]. However, such properties of stability are highly dependent on the nature of the operator matrix itself. As we have seen, PTS is not unconditionally stable in certain cases, for example, the one way wave equation with finite difference spatial discretization. In this chapter we discuss a class of implicit methods, Matrix Padé Time-stepping schemes(MPTS), which in contrast, time-steps the solution of differential equations in a matrix manner. Such matrix version of Padé approximations can be seen as early as in the book of G. A. Baker, Jr. and P. Graves-Morris[6], and has been developed by many mathematicians (see, for example [10, 11, 34]). MPTS can achieve higher stability than those classical methods and PTS schemes. In many cases, MPTS possess the unconditional stability property.

3.1 Matrix approximants by Padé approximations

Recall that, for a system of differential equations, $\dot{u} = A(t)u$, PTS schemes time-step the solution u by means of certain explicit algebraic manipulations on the Taylor polynomial (2.2), which is based on the vector component computation. Instead of that, the MPTS schemes time-step the solution u in a matrix manner.

3.1.1 Fixed Matrix Padé Approximation

Consider a general system of ordinary differential equations, which is in the form of

$$\dot{u} = A(t)u. \tag{3.1}$$

Here, $A(t)$ is a given, constant or explicitly dependent also on t or u or both, real or complex n-by-n matrix.

A solution vector u is sought which satisfies an initial condition

$$u(0) = u_0 \tag{3.2}$$

In principle, the solution is given by, for instance in the case where A is constant,

$$u(t) = e^{tA}u_0. \tag{3.3}$$

Here we focus on ODE cases only, since in practice many PDE problems are transformed into a system of ODEs; the results arising from ODE cases could also be applied to PDE cases.

By the definition of e^{tA}, it can be formally defined by the convergent power series, [23]

$$e^{tA} = I + tA + \frac{t^2A^2}{2!} + \frac{t^3A^3}{3!} + \cdots \tag{3.4}$$

By means of $[L, M]$ Padé approximation, e^{tA} is defined by [23]

$$e^{tA} = P_{LM}(tA) = [D_{LM}(tA)]^{-1}[N_{LM}(tA)] \tag{3.5}$$

where

$$N_{LM}(tA) = \sum_{j=0}^{L} \frac{(L+M-j)!L!}{(L+M)!j!(L-j)!}(tA)^j$$

and

$$D_{LM}(tA) = \sum_{j=0}^{M} \frac{(L+M-j)!M!}{(L+M)!j!(M-j)!}(-tA)^j$$

The nonsingularity of $D_{LM}(tA)$ is assured provided that L and M large enough, or that the eigenvalues of tA are negative. Zakian [35] and Wragg and Davies [33] considered the choice of L and M to obtain prescribed accuracy and efficiency. Diagonal ($L = M$) Padé approximants are preferred by their studies versus off diagonal approximations ($L \neq M$), though in this case it is mainly due to efficiency, rather than stability. Suppose $L < M$. About Mn^3 flops are required to evaluate $P_{LM}(tA)$, an approximation which has order $L + M$. However, the same amount of work is needed to compute $P_{MM}(tA)$ and this approximation has order $2M > L + M$. A similar argument can be applied to the superdiagonal approximations ($L > M$).

There are other reasons for favoring the diagonal Padé approximations. If all eigenvalues of tA are in the left half plane, then the computed approximants with $L > M$ tend to have larger rounding errors due to cancellation while the computed approximants with $L < M$ tend to have larger rounding errors due to badly conditioned matrices $D_{LM}(tA)$. Consequently, once again, we restrict

our focus to $[L, L]$ Padé approximations. This also ensures the comparisons with PTS are appropriate.

3.1.2 Taylor Matrix Padé approximants

In the context above, we have discussed the formal way to generate the $[L, L]$ matrix Padé approximants to the fundamental solution of differential equation with the linear operator A. However, consider the fundamental solutions of linear problems applying to evolving the solutions of nonlinear problems, the operator A normally is no longer fixed in general (usually depends on the value of u). As a result, the previous analysis becomes no longer applicable. For instance, suppose $\dot{u} = Au$, if A is constant, then obviously $\ddot{u} = A^2u$, which is suitable to apply the formula (3.4). However, if A is explicitly dependent on t, thus $\ddot{u} = (A^2 + A_t)u$ where A_t denotes the derivative of A with respect to t, which is the reason we cannot apply the formula (3.4) for Padé approximations. More arguments have been addressed by G. Xu [34].

There are already many papers that generalize the fixed matrix Padé approximation to general matrices in one way or another. The volumes by Xu [34] present more generalities and details on this subject. However, the purpose is to introduce the Matrix PTS schemes here, thus we omit those complicated discussions and concentrate on finding the matrix coefficients of Padé approximants to the exponential operators associated with the solutions of differential equations.

To get insight into the Taylor matrix Padé approximations, again, we consider a system of differential equations

$$\dot{u} = Au,$$

where A is a general matrix. It is well known [34] that the matrix e^{hA} has Padé approximants, for time-step h at time t,

$$\begin{aligned} e^{hA} &= P_{LL}(h) + O(h^{L+L+1}) \\ &= [I + B_1 h + \cdots + B_L h^L]^{-1}[A_0 + A_1 h + \cdots + A_L h^L] + O(h^{2L+1}), \end{aligned} \tag{3.6}$$

for some yet undetermined matrices $A_i, B_j, i, j = 1, ..., L$. The existence and uniqueness of $P_{LL}(h)$ have been proven by a number of studies. The study of Xu [34] and more recently a paper of A. Draux [13] were published on this subject. Suppose the matrix e^{hA} has Taylor series, in time-step h assumed $t = 0$,

$$e^{hA} = I + C_1 h + C_2 h^2 + C_3 h^3 + \cdots, \text{ for some matrices } C_1, C_2, C_3, \cdots \tag{3.7}$$

Then, with (3.6) and (3.7), we have

$$I + C_1 h + C_2 h^2 + C_3 h^3 + \cdots = [I + B_1 h + \cdots + B_L h^L]^{-1}[A_0 + A_1 h + \cdots + A_L h^L], \tag{3.8}$$

for some matrices $C_1, C_2, C_3, \cdots$, then formally the matrices $A_i, B_j, i, j = 1, ..., L$, can be determined as in the scalar case.

For example, if we consider the $[1, 1]$ Padé approximation, suppose

$$[I + C_1 h + C_2 h^2]u = [I + B_1 h]^{-1}[A_0 + A_1 h]u, \text{ for some matrices } B_1, A_0, A_1.$$

We have

$$
\begin{aligned}
& [I + B_1 h][I + C_1 h + C_2 h^2]u = [A_0 + A_1 h]u \\
\Rightarrow \quad & [I + [B_1 + C_1]h + [B_1 C_1 + C_2]h^2 + B_1 C_2 h^3]u = A_0 u + A_1 uh \\
\Rightarrow \quad & \begin{cases} A_0 = I \\ A_1 = B_1 + C_1 \\ B_1 C_1 + C_2 = 0 \Rightarrow B_1 C_1 = -C_2 \Rightarrow B_1 = -C_2 C_1^{-1} \end{cases} \qquad (3.9) \\
\Rightarrow \quad & A_1 = -C_2 C_1^{-1} + C_1 \\
\Rightarrow \quad & [I + C_1 h + C_2 h^2]u = [I - C_2 C_1^{-1} h]^{-1}[I + [-C_2 C_1^{-1} + C_1]h]u
\end{aligned}
$$

Which is the $[1,1]$ Padé approximant of e^{hA} with $\mathcal{O}(h^2)$ global accuracy, since it agrees up to $\mathcal{O}(h^2)$ with the Taylor expansion defined in (3.7).

Similar idea, for $[2,2]$ Padé approximation,

$$
\begin{aligned}
& [I + C_1 h + C_2 h^2 + C_3 h^3 + C_4 h^4]u \\
& = [I + B_1 h + B_2 h^2]^{-1}[A_0 + A_1 h + A_2 h^2]u, \text{ for some matrices } A_0, A_1, A_2, B_1, B_2.
\end{aligned}
$$

We have

$$
\begin{aligned}
& [I + B_1 h + B_2 h^2][I + C_1 h + C_2 h^2 + C_3 h^3 + C_4 h^4]u = [A_0 + A_1 h + A_2 h^2]u \\
\Rightarrow\ & [I + [B_1 + C_1]h + [B_1 C_1 + C_2 + B_2]h^2 + [C_3 + B_1 C_2 + B_2 C_1]h^3 \\
& + [C_4 + B_1 C_3 + B_2 C_2]h^4]u = A_0 u + A_1 u h + A_2 u h^2
\end{aligned}
$$

$$
\Rightarrow \begin{cases}
A_0 = I \\
A_1 = B_1 + C_1 \\
A_2 = C_2 + B_1 C_1 + B_2 \\
C_3 + B_1 C_2 + B_2 C_1 = 0 \\
C_4 + B_1 C_3 + B_2 C_2 = 0
\end{cases}
$$

$$
\Rightarrow \begin{cases}
B_1 = [C_4 C_2^{-1} - C_3 C_1^{-1}][C_2 C_1^{-1} - C_3 C_2^{-1}]^{-1} \\
B_2 = [C_3 C_2^{-1} - C_4 C_3^{-1}][C_2 C_3^{-1} - C_1 C_2^{-1}]^{-1}
\end{cases}
$$

$$
\Rightarrow \begin{cases}
A_1 = [C_4 C_2^{-1} - C_3 C_1^{-1}][C_2 C_1^{-1} - C_3 C_2^{-1}]^{-1} + C_1 \\
A_2 = C_2 + [C_4 C_2^{-1} - C_3 C_1^{-1}][C_2 C_1^{-1} - C_3 C_2^{-1}]^{-1} C_1 \\
\qquad + [C_3 C_2^{-1} - C_4 C_3^{-1}][C_2 C_3^{-1} - C_1 C_2^{-1}]^{-1}
\end{cases}
\tag{3.10}
$$

Thus, the $[2,2]$ Padé approximant is determined as above, which is $\mathcal{O}(h^4)$ accurate globally, since it agrees up to $\mathcal{O}(h^4)$ with the Taylor series defined in (3.7).

The same approach can be applied to obtain $[L, L]$ Padé approximations for any positive integer L.

3.1.3 Matrix PTS schemes

Thus given a system of ODEs $\dot{u} = Au$, and given a solution u at time t, we can generate the approximate solution at $t + h$ by expanding the solution u via its Taylor polynomial in time-step h

$$u(t+h) \approx \sum_{j=0} C_j^t h^j \tag{3.11}$$

The MPTS scheme time-steps the solution u by means of $[L, L]$ matrix Padé approximation on the Taylor polynomial (3.11), denoted by $P_{L,L}^t(h)$, where

$$\begin{aligned} P_{L,L}^t(h) &= [B^t(h)]^{-1} A^t(h) \\ &= [I + B_1^t h + B_2^t h^2 + \cdots + B_L^t h^L]^{-1} [A_0^t + A_1^t h + \cdots + A_L^t h^L] \end{aligned} \tag{3.12}$$

provided it exists. Then, letting $t_j = jh$, the j-th time-step (j=1,2,...), $[L, L]$ MPTS scheme time-steps the solution by

$$u_{j+1} = P_{L,L}^{t_j}(h)$$

unless one of the following situations occur:

1. The matrix Padé approximant $P_{L,L}^{t_j}$ does not exist, or
2. The matrix Padé approximant $P_{L,L}^{t_j}$ exists but the condition

$$|P_{L,L}^{t_j}(h)| < K|u_j| \tag{3.13}$$

is violated - where K is an appropriately large constant, and problem dependent. This situation may occur when, for instance, the matrix $B^t(h)$ is singular or near singular. In other words, the absolute value of the determinant of $B^t(h) < \epsilon$, where ϵ is a small constant.

Thus the condition 2 above can be reverted to

2′. The matrix Padé approxomant $P^{l_j}_{L,L}$ exists but, the condition

$$|det(B^l(h))| > \epsilon \tag{3.14}$$

is violated, where $det(B)$ denotes the determinant of a matrix B.

Note that the selection of constant ϵ can be problem dependent, in practice, it has been chosen from a range of $(10^{-17} \sim 10^{-13})$, which can avoid most of such condition violations. However, in our numerical experiment, we found that it's always possible that there is no adequate ϵ exists in order to eliminate the voilation to 2 or 2′. In this case, a condition control statement and an appropriate truncated Taylor polynomial should be employed in the schemes.

3.2 Stability of MPTS

Once again, we study the stability properties of MPTS$[L, L]$ scheme based on the consideration of their A-stability, and we claim that

Theorem 2. *Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ be the eigenvalues of an $k \times k$ matrix A, $\lambda_i \in \boldsymbol{R}$ and $\lambda_i \leq 0, i = 1, 2, \ldots, k$. Then, for any positive integer L, the MPTS$[L, L]$ scheme for the system of ODEs $\dot{u} = Au$, where A is a diagonalizable matrix, is A-stable for all step sizes $h > 0$. In addition, if $\forall m\ \lambda_m < 0$, then the MPTS$[L, L]$ numerical solution u_j for this system tends to zero as $j \longrightarrow \infty$.*

Proof. Consider the matrix A drawn from a system of k ODEs $\dot{u} = Au$.

A can be rewritten as

$$A = VDV^{-1},$$

where V are the matrix of eigenvectors of A, and D is a diagonal matrix whose diagonal elements are the eigenvalues of A. Let

$$D = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & O & \\ & & \ddots & \\ & O & & \lambda_{k-1} \\ & & & & \lambda_k \end{bmatrix},$$

By MPTS$[L, L]$ scheme with a step size $h > 0$, $u_{j+1} = P(h)u_j$, where

$$\begin{aligned} P(h) &= \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(-Ah)^j\right)^{-1} \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(Ah)^j\right) \\ &= \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(-V(Dh)V^{-1})^j\right)^{-1} \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(V(Dh)V^{-1})^j\right) \\ &= \left(V\left[\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(-Dh)^j\right]V^{-1}\right)^{-1} \left(V\left[\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(Dh)^j\right]V^{-1}\right) \\ &= V\left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(-Dh)^j\right)^{-1} \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(Dh)^j\right)V^{-1} \\ &= VBV^{-1}, \end{aligned}$$

where

$$\begin{aligned} B &= \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(-Dh)^j\right)^{-1} \left(\sum_{j=0}^{L} \tfrac{(2L-j)!L!}{(2L)!j!(L-j)!}(Dh)^j\right) \\ &= \begin{bmatrix} f(\lambda_1) & & & \\ & f(\lambda_2) & O & \\ & & \ddots & \\ & O & & f(\lambda_{k-1}) \\ & & & & f(\lambda_k) \end{bmatrix}, \end{aligned}$$

with $f(\lambda_i)$ defined as

$$f(\lambda_i) = \frac{\sum_{j=0}^{L} \frac{(2L-j)!L!}{(2L)!j!(L-j)!}(\lambda_i h)^j}{\sum_{j=0}^{L} \frac{(2L-j)!L!}{(2L)!j!(L-j)!}(-\lambda_i h)^j}, \quad i = 1, 2, \ldots, k$$

Note that, since the eigenvalues of A are λ_i, then the eigenvalues of P are $f(\lambda_i)$. Thus for stability, we need $|f(\lambda_i)| \leq 1 \quad \forall i$. Since $\lambda_i \leq 0$ for all $i = 1, 2, \ldots, k$, $(\lambda_i h)^j \leq (-\lambda_i h)^j$ for any positive integer j, therefore,

$$\left| \sum_{j=0}^{L} \frac{(2L-j)!L!}{(2L)!j!(L-j)!}(\lambda_i h)^j \right| \leq \left| \sum_{j=0}^{L} \frac{(2L-j)!L!}{(2L)!j!(L-j)!}(-\lambda_i h)^j \right|, \text{ for all } i = 1, 2, \ldots, k$$

which implies $|f(\lambda_i)| \leq 1 \quad \forall i$. Hence MPTS$[L, L]$ is stable for any step-size $h > 0$. And on top of that, if $\forall m$ such that $\lambda_m < 0$, then $|f(\lambda_i)| < 1 \quad \forall i \Rightarrow |P| < 1 \Rightarrow |P^j| \to 0$ as $j \to \infty$, thus $|u_{j+1}| = |Pu_j| = |P||u_j| \to 0$ as $j \to \infty \Rightarrow u_{j+1} \to 0$ as $j \to \infty$, as claimed. □

Remark 1. *Theorem 2 holds for operators whose eigenvalues are complex as well, if all eigenvalues of the operator have nonpositive real part. In such cases, $P^j \to 0$ as $j \to \infty$. Therefore, the solution given by MPTS stays stable.*

In the theorem above, we have shown that MPTS schemes possess A-stability property when A is diagonalizable. On the other hand, in case of a non-diagonalizable matrix A, by using the techniques of Jordan decomposition [18], we can always construct the Jordan form of this matrix (since A is a square matrix) which is of block diagonalizable form. By the idea in the following chapters, this block diagonalizable form is an ideal structure that our block splitting method is able to overcome.

Chapter 4

Block Splitting of Exponential Operators

4.1 Introduction

For a system of differential equations (3.1), a solution, or an approximate solution can be obtained by (3.3), satisfying a certain initial condition (3.2). However, computing the matrix exponential demands high processing capabilities. Even with modern well equipped computers, such computations may lead to a huge amount of computing time. Despite the fact that MPTS schemes possess strong stability properties and desired convergence, in each time step t_j, they involve evaluating the solution u_j of a linear system problem, whose dimension is that of the ODE system. For example, in the case of PDE problems, when the spatial difference is small, the computation of such evaluation can suffer the same difficulty as that of matrix exponential.

In order to avoid such difficulty, while the high order accuracy is retained, a concept of operator splitting has been and continues to be used widely for many types of evolution equations. More specifically, splitting methods for time-dependent partial differential equations have been most frequently studied in the context of spatial splittings, as in the approximate factorization techniques for efficiently implementing implicit algorithms [14, 20, 26, 28]. Some attention has also been given to splitting or fractional step methods for problems where the differential operator is split up into pieces corresponding to different physical processes which are most naturelly handled by different techniques. This has been done, for example, with the convection-diffusion equation and the nonlinear Schrödinger equation [7, 17].

More generally, a splitting method may be useful any time one is faced with a problem like (3.1), where A is some differential operator, which may be separated as

$$A = A_1 + A_2 \tag{4.1}$$

such that the problems

$$\dot{u} = A_1 u \tag{4.2}$$

and

$$\dot{u} = A_2 u \tag{4.3}$$

are each easier to solve than the original problem. By alternating between solving (4.2) and (4.3) we hope to compute a satisfactory solution to (3.1).

Note that, in (4.1), we just give an example of the simplest form by such splitting, where the operator is split into 2 sub-operators. In fact, the number of sub-operators can be varied, and of course, depends on the nature of the original operator, the order of splitting accuracy, the efficiency of implementation, etc.

In this context, an introduction of operator splitting is given by a general M sub-operators arising from a general linear PDE problem, with a m-th order accuracy splitting. The results, however, also apply to discretizations of ODEs [14].

4.2 The m-th order operator splitting

Let us consider an operator A of the system of differential equations (3.1), given a certain step size h and j-th step solution u_j, it has its $(j+1)^{th}$ time step solution $u_{j+1} = e^{hA}u_j$ satisfying a certain initial condition $u(x,0) = u_0$. As is well known [14, 26], by mean of operator splitting, the operator matrix A can be split into M blocks, $A_1, A_2, ..., A_M$ for non-commutable sub-operators $\{A_j\}$, such that

$$e^{hA} = e^{h(s_1A_1+s_2A_2+\cdots+s_MA_M)} + \mathcal{O}(h^{m+1}) = S_m(h) \tag{4.4}$$

with splitting factors $\{s_j\}$ and finite m associated with M.

The first and second order operator splits are well known and easy to obtain. For higher orders, the simplest and most efficient way to split the operator is to find a systematic series of approximants (4.4) by a recursive fractal decomposition [29]. Consider the following $(m-1)^{th}$ approximant ($m \geq 3$):

$$exp\left(h\sum_{j=1}^{M} A_j\right) = S_{m-1}(h) + \mathcal{O}(h^m) \tag{4.5}$$

Then, the m^{th} approximant $S_m(h)$ is constructed as follows [29]:

$$S_m(h) = \prod_{j=0}^{r} S_{m-1}(s_{m,j}h), \tag{4.6}$$

for $r \geq 2$, where the parameters $\{s_{m,j}\}$ are the solutions of the following decomposition condition that

$$\sum_{j=1}^{r} s_{m,j}^{m} = 0, \text{ with } \sum_{j=0}^{r} s_{m,j} = 1. \tag{4.7}$$

It's easy to verify (4.6) and (4.7) from the following identity:

$$exp\left(h\sum_{k=1}^{M} A_k\right) = \prod_{j=1}^{r} exp\left(s_{m,j}h\sum_{k=1}^{M} A_k\right). \tag{4.8}$$

Then, we substitute the $(m-1)^{th}$ approximant $S_{m-1}(s_{m,j}h)$ in each factor of (4.8). The decomposition condition (4.7) is derived both from the requirement that the sum of the uncontrollable m^{th}-order terms in (4.8)

$$h^m \left(\sum_{j=1}^{r} s_{m,j}^{m}\right)\left(\sum_{k=1}^{M} A_k\right)^m. \tag{4.9}$$

should vanish, and from the requirement that the corresponding sum of the m-th order terms in each $S_{m-1}(s_{m,j}h)$ should also vanish. In order to study the latter condition explicitly, we write the m^{th}-order term of $S_{m-1}(h)$ as

$$[S_{m-1}(h)]_m = h^m P_m(\{A_j\}). \tag{4.10}$$

Then, the sum of the m^{th}-order terms in each factor of the right-hand side of (4.8) is given by

$$h^m \left(\sum_{j=1}^{r} s_{m,j}^{m}\right) P_m(\{A_j\}). \tag{4.11}$$

Thus we obtain the two uncontrollable expressions (4.9) and (4.11) vanish under the single common condition

$$\sum_{j=1}^{r} s_{m,j}^{m} = 0. \tag{4.12}$$

With this general discussion, now we start to study the simplest two-block-splitting, where

$$e^{h(A)} = e^{h(A_1+A_2)}. \tag{4.13}$$

As it is well known [26], the first order (in h) block splitting is given by

$$S_1(h) = e^{hA_1}e^{hA_2}. \tag{4.14}$$

The second-order block splitting is given by the following symmetric product

$$S_2(h) = e^{\frac{h}{2}A_1}e^{hA_2}e^{\frac{h}{2}A_1}. \tag{4.15}$$

For the case $m = 3$ in (4.4), let we start from the following identity

$$e^{h(A_1+A_2)} = e^{sh(A_1+A_2)}e^{(1-2s)h(A_1+A_2)}e^{sh(A_1+A_2)}. \tag{4.16}$$

which gives the third-order symmetric approximant

$$S_3(h) = S_2(sh)S_2((1-2s)h)S_2(sh), \tag{4.17}$$

where the parameter s, by what we have shown in (4.7), is given by the real solution of the equation

$$2s^3 + (1-2s)^3 = 0 \qquad \Rightarrow s = \frac{1}{2-3\sqrt{2}} = 1.3512... \tag{4.18}$$

Thus the block splitting of third order (in h)is given explicitly by

$$S_3(h) = e^{\frac{s}{2}hA_1}e^{shA_2}e^{\frac{1-s}{2}hA_1}e^{(1-2s)hA_2}e^{\frac{1-s}{2}hA_1}e^{shA_2}e^{\frac{s}{2}hA_1}, \tag{4.19}$$

with s in (4.18). An equivalence theorem between the $(2m-1)$-th and $2m$-th ($m \geq 2$) approximants has been established and proved by M. Suzuki [29]. In our nomenclature, we can state

Theorem 3. *We assume that the original operator $A(h)$ with a parameter h is symmetric in the sense that*

$$A(h)A(-h) = 1; \quad A(0) = 1,$$

and for it we construct, in general, a symmetric $(2m-1)$*-th order approximant* $S_{2m-1}(h)$*, where* $m \geq 2$*, namely,*

$$A(h) = S_{2m-1}(h) + \mathcal{O}(h^{2m}),$$

where

$$S_{2m-1}(h)S_{2m-1}(-h) = 1.$$

Then, $S_{2m-1}(h)$ *is also correct up to the order of* h^{2m}*, namely,*

$$S_{2m-1}(h) = S_{2m}(h).$$

Therefore with Theorem 3, we can obtain the fourth-order approximant $S_4(h)$ as

$$S_4(h) = S_3(h).$$

In general, the $(2m-1)$-th and $2m$-th $(m \geq 2)$ approximants, $S_{2m-1}(h)$ and $S_{2m}(h)$, are determined recursively as [29]

$$S_{2m-1}(h) = S_{2m}(h) = S_{2m-3}(s_m h)S_{2m-3}((1-2s_m)h)S_{2m-3}(s_m h), \tag{4.20}$$

where

$$s_m = (2 - 2^{1/(2m-1)})^{-1}. \tag{4.21}$$

With the basic introductions above, a practical scheme of real decomposition was derived for real $\{h\}$. For this purpose, consider the following symmetric real decomposition for the exponential operator

$$A(h) \equiv e^{h(A_1+A_2+\cdots+A_M)} = S_{2m}(h) + \mathcal{O}(h^{2m+1}). \tag{4.22}$$

From Theorem 3, we have the recursion formula

$$S_{2m}(h) = S_{2m-1}(h) = [S_{2m-3}(s_m h)]^2 S_{2m-3}((1-4s_m)h)[S_{2m-3}(s_m h)]^2$$

with, for instance, the first- and second- order symmetrized splitting

$$\begin{aligned} S_1(h) &\equiv e^{hA_1}e^{hA_2}\cdots e^{hA_M}\cdots e^{hA_2}e^{hA_1} \\ S_2(h) &\equiv e^{\frac{h}{2}A_1}e^{\frac{h}{2}A_2}\cdots e^{\frac{h}{2}A_{M-1}}e^{hA_M}e^{\frac{h}{2}A_{M-1}}\cdots e^{\frac{h}{2}A_2}e^{\frac{h}{2}A_1}, \end{aligned} \tag{4.23}$$

where the parameter s_m is the real solution of the equation (4.21). Consequently, the parameters s_j in (4.4) for the $2m$-th order approximant are given by the product of some combinations of

$$s_2, s_3, \cdots s_m, 1-4s_2, 1-4s_3, \ldots, 1-4s_m.$$

Therefore, we have

$$\lim_{m\to\infty} s_j = 0, \quad \text{for all } j.$$

4.3 Convergence of block splittings of exponential operators

In the present section, we investigate and state related results about the convergence of some systematic series of decompositions of exponential operators defined as $exp[(A_1 + A_2 + \cdots + A_M)]$ for non-commutable operators $\{A_j\}$. In this thesis we mean bounded linear or nonlinear operator arising from a system of differential equations.

In the previous section, we have shown that, in general, the m-th order exponential decomposition $S_m(h)$ is given in the form

$$S_m(h) = e^{hs_{11}A_1}e^{hs_{12}A_2}\ldots e^{hs_{1M}A_M}e^{hs_{21}A_1}e^{hs_{22}A_2}\ldots e^{hs_{2M}A_M}\ldots, \tag{4.24}$$

with some appropriate parameters $\{s_{ij}\}$ determined by the requirement that

$$e^{h(A_1+A_2+\cdots+A_M)} = S_m(h) + \mathcal{O}(h^{m+1}), \tag{4.25}$$

where for example, the first-order decomposition $S(h)$ is given by

$$S_1(h) = e^{hA_1}e^{hA_2}\dots e^{hA_{M-1}}e^{hA_M} \tag{4.26}$$

i.e

$$e^{h(A_1+A_2+\dots A_{M-1}+A_M)} = S_1(h) + \mathcal{O}(h^2), \tag{4.27}$$

and the second-order symmetric decomposition is given by

$$S_2(h) = e^{\frac{h}{2}A_1}\dots e^{\frac{h}{2}A_{M-1}}e^{hA_M}e^{\frac{h}{2}A_{M-1}}\dots e^{\frac{h}{2}A_1}, \tag{4.28}$$

i.e.

$$e^{h(A_1+A_2+\dots A_{M-1}+A_M)} = S_2(h) + \mathcal{O}(h^3), \tag{4.29}$$

It is more convenient to get the mth order approximant $S_m(h)$ of $exp[h(A_1 + A_2 + \cdots + A_M)]$ as a decomposition in terms of n-th order approximant $S_n(h)$ as

$$S_m(h) = S_n(s_1h)S_n(s_2h)\dots S_n(s_kh) \tag{4.30}$$

for some appropriate parameters $\{s_j\}$ which satisfy the condition

$$s_1 + s_2 + \cdots + s_k = 1,$$

and some other relations [30]. Here k depends on m and k tends to the infinity as $m \mapsto \infty$.

A systematic scheme to derive higher-order decompositions even up to infinite order is given by the following recursive method as we stated in the previous section

$$S_{2m}(h) = S_{2m-2}(s_{m,1}h)S_{2m-2}(s_{m,2}h)\cdots S_{2m-2}(s_{m,k}h) \tag{4.31}$$

with

$$s_{m,1} + s_{m,2} + \cdots + s_{m,k} = 1 \quad \text{and} \quad s_{m,1}^{2m-1} + s_{m,2}^{2m-1} + \cdots + s_{m,k}^{2m-1} = 0.$$

Now the problem is to study the convergence of the general m-th order decomposition $S_m(h)$ in the limit $m \to \infty$.

In general, the decomposition with parameters $\{s_{m,j}\}$ is called "fractal", when the parameters $\{s_{m,j}\}$ satisfy the following conditions:

1. $s_{m,1} + s_{m,2} + \cdots + s_{m,k} = 1$,
2. $|s_{m,j} + s_{m,j+1} + \cdots + s_{m,k}|$ is bounded uniformly for both m and j, and
3. $\sum_{j=0}^{k} |s_{m,j}| \to \infty$ as $m \to \infty$

Here, the parameter k depends on the index m, namely $k = k(m)$. We also say that the m-th order decomposition is "of index m".

In order to study the convergence of the decomposition of (4.30) or more general non-uniform decompositions in the limit $m \to \infty$, we also need the following definition of an "approximant of index m" [28].

Definition 1. *A family of operators $S_m^{(j)}(h)$ depending on $h \in \mathbb{C}$ such that*

$$||S_m^{(j)}(h) - e^{hA}|| \leq K_m |h|^{m+1}$$

where $A = A_1 + A_2 + \cdots + A_M$, holds uniformly for any j for $|h| < \epsilon$, with some $\epsilon > 0$, with a positive number m and with some positive constant K_m, all independent of j, is called an approximant of index m.

Then, the following theorem is well established [28].

Theorem 4. *Let $S_m^{(j)}(h)$ be approximants of index m for the exponential operator $exp(hA) \equiv exp[h(A_1 + A_2 + \cdots + A_M)]$. A systematic series of approximants*

$\{S_m(h)\}$ for $exp(hA)$ constructed by the ordered product

$$S_m(h) = S_n^{(1)}(s_{m,1}h)S_n^{(2)}(s_{m,2}h)\cdots S_n^{(m)}(s_{m,k}h) \qquad (i)$$

converges to $exp(hA)$, namely

$$\lim_{m\to\infty} ||S_m(h) - e^{hA}|| = 0, \ \ i.e. \ \ \lim_{m\to\infty} S_m(h) = e^{hA} \qquad (ii)$$

for all $h \in \mathbb{C}$ under the condition that

$$\lim_{m\to\infty} \sum_{j=1}^{k(m)} |s_{m,j}^{n+1}| = 0 \qquad (iii)$$

together with the conditions 1 and 2 above, the limit(ii) is uniform provided $|h| < \delta$ for any positive number δ.

Conversely, if $S_n^{(1)}(h) = \cdots = S_n^{(n)}(h) = S_n(h)$ for a strictly n-th order $S_n(h)$ and $S_m(h)$ converges uniformly to e^{hA} for any operators $\{A_j\}$, then

$$\lim_{m\to\infty} \sum_{j=1}^{k(m)} s_{m,j}^{n+1} = 0$$

Remark 2. This theorem can be applied to fractal decompositions in which $\sum_j |s_{m,j}|$ diverges in the limit $m \to \infty$. It is easy to confirm the condition (iii) above for fractal decompositions [28].

Remark 3. The parameter k in conditions (i) and (iii) increases as m increases, namely $k = k(m) \to \infty$ as $m \to \infty$. The parameters $\{s_{m,j}\}$ go to zero as m goes to infinity, as is required from condition (iii). However, it does not necessarily imply the boundedness of the summation $\sum_j |s_{m,j}|$.

Corollary 1. *If we construct $S_m(h)$ as*

$$S_m(h) = S^{(1)}(s_{m,1}x)S^{(2)}(s_{m,2}h)\ldots S^{(k)}(s_{m,k}h) \qquad (iv)$$

with $S^{(j)}(h)$ of the form (4.26), namely of first order, then

$$\lim_{m\to\infty} S_m(h) = e^{hA} \tag{v}$$

under the condition that

$$\lim_{m\to\infty} \sum_{j=1}^{k(m)} |s_{m,j}|^2 = 0,$$

together with the condition 2 above.

Conversely, if $S^{(1)}(h) = \cdots = S^{(n)}(h) = S(h)$ and limit (iv) holds, then

$$\lim_{m\to\infty} \sum_{j=1}^{k(m)} s_{m,j}^2 = 0. \tag{vi}$$

Furthermore, for the real decomposition (i.e., for real $\{s_{m,j}\}$) satisfying condition 2 and limit (vi) is a necessary and sufficient condition for the convergence (iv).

Corollary 2. *For the decomposition*

$$S_{2m}(h) = S(s_{m,1}h)S(s_{m,2}h)\ldots S(s_{m,k}h), \tag{vii}$$

we have

$$\lim_{m\to\infty} S_{2m}(h) = e^{h(A_1+\cdots+A_M)} \tag{viii}$$

under the condition that

$$\lim_{m\to\infty} \sum_{j=1}^{k(m)} |s_{m,j}|^3 = 0,$$

together with (ii).

Conversely, if (viii) holds, then

$$\lim_{m\to\infty} \sum_{j=1}^{k(m)} s_{m,j}^3 = 0,$$

4.4 Necessity of negative coefficients for splitting schemes of order greater than two

Numerical schemes of order $m > 2$ based on the composition (4.31) have been successfully applied for solving a large number of problems [29, 30], including certain partial differential equations [7, 17, 20]. In fact, splitting methods are frequently used in celestial mechanics, quantum mechanics, molecular dynamics, accelerator physics and, in general, for numerically solving Hamiltonian dynamical systems, Poisson systems and reversible differential equations [21]. It has been noticed, however, that some of the coefficients in (4.31) are negative for $m > 2$. In other words, the methods in higher order always involve stepping backwards in time. The existence of backward fractional time steps in the composition method (4.31) is in fact unavoidable, and can be established as the following theorem [14, 26]:

Theorem 5. *If m is a positive integer such that $m \geq 3$, then there are no composition methods of the form* (4.31) *and finite m with all the coefficients $\{s_{m,j}\}$ being positive.*

This constitutes a problem when the differential equation is defined in a semigroup, as arises sometimes in applications, since then the method can only be conditionally stable [21]. Also schemes with negative coefficients may not be well-posed when applied to PDEs involving unbounded operators, or more crucially, the parabolic PDEs.

Chapter 5

Block Splitting PTS Schemes

5.1 Introduction

Operator splitting methods are used in many circumstances for solving ordinary and partial differential equations. Advances have been made in the application of these methods to equations with large sparse eigenvalue problems. However, in some certain cases, for instance the parabolic equations, higher order methods have been less used partly due to a result by Sheng in a 1989 [26], Sheng showed that higher order (order greater than two) operator splitting methods must contain operators which integrate backwards in time. This result has also been found by others [30]. Sheng also showed that a certain class of split operators containing sums of product terms, some terms having backward time evolution, were unstable in integrating a split heat equation. Therefore, the conclusion for this class of split operators was that they were unstable for higher order methods.

In the our study, a method of operator splitting via Padé approximation leads to the Block Splitting Padé Time-stepping (BSPTS) schemes which are introduced to numerically evolve systems of ordinary differential equations. Once again, since in practice many PDE problems can be approximated into a system of ODEs to evaluate, the results arising from ODE cases could also be applied to PDE problems.

The new schemes have the following features:

1. Based on the nature of the original operator, the size of blocks can be chosen to satisfy a diverse range of approach goals, in terms of efficiency, stability and accuracy;
2. A-stability over a wide range of time-steps for a variety of ODE problems;
3. They can be stable in evolving parabolic PDEs for higher orders (orders greater than 2);
4. The schemes are simple and easy to implement.

In short, the present method is a natural extension of the operator splitting method, and a generation of PTS and MPTS schemes, acting as an 'intermediate' approach in terms of stability and computational expense per step. Comparison of the present scheme with the PTS and MPTS schemes shows that the BSPTS improves the stability of PTS on certain cases, with only limited extra per-step computational cost, and it is more efficient than MPTS.

5.2 Block splitting Padé approximants

Again, let us consider the operator A arising from the system of differential equations (2.1), $\dot{u} = Au$ with initial value $u(t = 0) = u_0$, and

$$A = s_1 A_1 + s_2 A_2 + \cdots + s_M A_M. \tag{5.1}$$

In the previous chapter we have shown that, with $h = \Delta t > 0$, the m-th order operator splitting is given by

$$\begin{aligned}
S_1(h) &= e^{hs_1A_1}e^{hs_2A_2}\cdots e^{hs_MA_M} && m = 1\\
S_2(h) &= e^{\frac{h}{2}s_1A_1}\cdots e^{\frac{h}{2}s_{M-1}A_{M-1}}e^{hs_MA_M}e^{\frac{h}{2}s_{M-1}A_{M-1}}\cdots e^{\frac{h}{2}s_1A_1} && m = 2\\
S_4(h) &= S_2(\omega h)S_2((1-2\omega)h)S_2(\omega h) && m = 4\\
&\quad \text{where } 2\omega^3 + (1-2\omega)^3 = 0 &&\\
\cdots &\quad \cdots && \cdots
\end{aligned} \tag{5.2}$$

Note that, all higher order splittings (order greater than 2) are combinations of lower order splittings, and can be finally presented in terms of the S_2. Therefore, without loss of generality, in what follows we discuss the Padé approximants of S_2 only.

Let $P^n(hA)$ denote the n-th order of Padé approximant of a certain operator A with a given step-size $h > 0$, and $P_m^n(hA)$ denotes the m-order splitting of $P^n(hA)$. Then

$$\begin{aligned}
S_2(h) &= P_2^n(hA) + \mathcal{O}(h^{n+1})\\
&= P^n(s_1\tfrac{h}{2}A_1)\cdots P^n(s_{M-1}\tfrac{h}{2}A_{M-1})P^n(s_MhA_M)P^n(s_{M-1}\tfrac{h}{2}A_{M-1})\cdots P^n(s_1\tfrac{h}{2}A_1)\\
&\quad +\mathcal{O}(h^{n+1})
\end{aligned} \tag{5.3}$$

Now an appropriate form of (3.5) or (3.8) can be applied to express the n-th order Padé approximants of $P^n(s_k\frac{h}{2}A_k)$, $k = 1, 2, \ldots, M-1$ or $P^n(s_M hA_M)$.

Here we present a simple example to gain an insight on the block splitting Padé approximation. Consider a tridiagonal matrix

$$A = \begin{bmatrix} a & b & 0 & 0 \\ c & d & e & 0 \\ 0 & f & g & x \\ 0 & 0 & y & z \end{bmatrix}, \tag{5.4}$$

where $a, b, c, d, e, f, g, x, y, z$ are real constants. If we let

$$A_1 = \begin{bmatrix} a & b & 0 & 0 \\ c & \frac{d}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{d}{2} & e & 0 \\ 0 & f & \frac{g}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } A_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{g}{2} & x \\ 0 & 0 & y & z \end{bmatrix} \tag{5.5}$$

Clearly, $A = A_1 + A_2 + A_3$. Thus, with $h > 0$, the second order of exponential splitting S_2 for the matrix A is

$$S_2(h) = e^{\frac{h}{2}A_1} e^{\frac{h}{2}A_2} e^{hA_3} e^{\frac{h}{2}A_2} e^{\frac{h}{2}A_1}. \tag{5.6}$$

If we consider a second order Padé approximations of S_2, i.e. $n = 2$, which is its [1,1] Padé approximant P^2, can be generated by applying the matrix Padé approximation (3.5), such that

$$\begin{aligned} P^2(\tfrac{h}{2}A_1) &= (I - \tfrac{A_1}{2}\tfrac{h}{2})^{-1}(I + \tfrac{A_1}{2}\tfrac{h}{2}), \\ P^2(\tfrac{h}{2}A_2) &= (I - \tfrac{A_2}{2}\tfrac{h}{2})^{-1}(I + \tfrac{A_2}{2}\tfrac{h}{2}), \\ P^2(hA_3) &= (I - \tfrac{A_3}{2}h)^{-1}(I + \tfrac{A_3}{2}h). \end{aligned} \tag{5.7}$$

Therefore, with (5.6) and (5.7), the 2nd order block splitting Padé approximant for matrix A

$$\begin{aligned} P_2^2(hA) &= (I - \tfrac{A_1}{2}\tfrac{h}{2})^{-1}(I + \tfrac{A_1}{2}\tfrac{h}{2})(I - \tfrac{A_2}{2}\tfrac{h}{2})^{-1}(I + \tfrac{A_2}{2}\tfrac{h}{2})\cdot \\ &\quad (I - \tfrac{A_3}{2}h)^{-1}(I + \tfrac{A_3}{2}h)(I - \tfrac{A_2}{2}\tfrac{h}{2})^{-1}(I + \tfrac{A_2}{2}\tfrac{h}{2})(I - \tfrac{A_1}{2}\tfrac{h}{2})^{-1}(I + \tfrac{A_1}{2}\tfrac{h}{2}). \end{aligned} \tag{5.8}$$

Remark 4. *For the block splitting Padé approximant P_m^n, the overall order is $r = min(m+1, n+1)$, i.e.*

$$e^{hA} = P_m^n(hA) + \mathcal{O}(h^{r+1}), \qquad \textit{where } r = min(m, n).$$

This remark can be easily seen, since

$$\mathcal{O}(h^{r+1}) = \mathcal{O}(h^{m+1}) + \mathcal{O}(h^{n+1}) = \mathcal{O}(h^{min(m,n)+1}).$$

According to the Remark 4, for the r-th order block splitting Padé approximant of e^{hA}, the r-order of block splitting S_r with its $[N, N]$ (where $2N \geq r$) Padé approximant has to be employed. Henceforce, in practice we should always choose $n = m$ for efficiency, and in this thesis we discuss P_n^n only.

5.3 Optimal block splitting Padé approximants

In fact, the way to split the operator is not unique. Any splitting blocks $\{A_i\}$ satisfying (5.1) can be considered, where an optimal splitting, in terms of stability, efficiency and/or other factors, for example the difficulties of implementation, is suggested for certain problems. For instance, for the previous example, an

alternative splitting read as

$$A_1 = \begin{bmatrix} a & b & 0 & 0 \\ c & \frac{d}{2} & \frac{e}{2} & 0 \\ 0 & \frac{f}{2} & \frac{g}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{d}{2} & \frac{e}{2} & 0 \\ 0 & \frac{f}{2} & \frac{g}{2} & x \\ 0 & 0 & y & z \end{bmatrix}, \tag{5.9}$$

and the associated 2nd order Padé approximation for this splitting can be

$$P_2^2(hA) = (I - \frac{A_1}{2}\frac{h}{2})^{-1}(I + \frac{A_1}{2}\frac{h}{2})(I - \frac{A_2}{2}h)^{-1}(I + \frac{A_2}{2}h)(I - \frac{A_1}{2}\frac{h}{2})^{-1}(I + \frac{A_1}{2}\frac{h}{2}). \tag{5.10}$$

As we can see from this example, the operator A is split into several suboperators $\{A_i\}$ with far less nonzero elements than A in relaxation manner to simplify the system we originally have. However, the nature of suboperators can also offer us another opportunity to ameliorate the algebraic computation. For instance, A_1 and A_2 in the last example, each of them has at least one zero-row and one zero-column which in fact reduce the dimension of the suboperator; consequently, it can be more computationally efficient when such operators are evaluated by certain algebraical method than that of the original operator.

While in general for a given order of accuracy several splittings could be considered for size and number of suboperators, our experiements suggest that the smaller is the size of suboperators, the more efficient is the computation per step; consequently, henceforth we restrict our focus to block splitting Padé approximants with the smallest suboperator size which provides the A-stability of the overall scheme, namely ***optimal block splitting Padé approximant***.

5.4 Block Splitting Padé Time-stepping

Our Block Splitting Padé Time-stepping (BSPTS) schemes evolves the solution u of (2.1) with suboperators $\{A_i\}$ satisfying (5.1) by means of the block splitting Padé approximant for each suboperators. Therefore, given an intergers N, and the time-step h at time t, let $n = 2N$, we denote $P_n^{n,t}(hA)$ the block splitting Padé approximant with order n, which can be derived by the procedures described in the last section. Then calling t_j the $j-$th time-step ($j = 1, 2, \dots$), the n-th order Block Splitting Padé Time-stepping scheme, S_n BSPTS$[N, N]$ (BSPTS$[N, N]$ by short) time-steps the solution according to the prescription

$$u_{j+1} = P_n^{n,t_j}(hA)u_j, \quad u(0) = u_0, \tag{5.11}$$

again, unless one of the following situations occur:

1. The block splitting Padé approximant $P_n^{n,t}(hA)$ does not exist for one or more suboperators, or

2. The block splitting Padé approximant $P_n^{n,t}(hA)$ exists for all suboperators, but the condition

$$|P_n^{n,t}(hA)| < K|u_j| \tag{5.12}$$

for some suboperators is violated - where K is an appropriately large constant, and problem dependent. This situation may occur when a certain suboperator A_i is singular or near singular. In other words, the absolute value of the determinant of $A_i < \epsilon$, where ϵ is a small constant.

Thus the condition 2 above can also be reverted to

2′. The block splitting Padé approximant $P_n^{n,t}(hA)$ exists for all suboperators, but the condition

$$|det(A_i)| > \epsilon \text{ for certain suboperators } A_i's, \text{ and constant } \epsilon \qquad (5.13)$$

is violated. In this case, a condition control statement and an appropriate truncated matrix-type Taylor polynomial should be considered in the algorithm.

Despite the desirable situation that (5.13) holds for every suboperator, the method may still remain valid when it is violated only for a certain number of suboperators. In this thesis we discuss the earlier case in 2′ only, the more challenging latter case will be a part of future work from this study.

Chapter 6

Stability of BSPTS Schemes

In this chapter, we study the stability of BSPTS schemes assuming, for simplicity, that the operator A is constant. Again, the stability property is drawn from consideration of its A-stability as well; for the stability in the following text we also mean the A-stability.

In practice, by the BSPTS schemes, we can consider the solution u^j at time t_j as

$$u_j = u_j^1 + u_j^2 + \cdots + u_j^M, \tag{6.1}$$

where u_j^i is the partial solution associated with the suboperator A_i at time t_j.

Definition 2. *Given a set of splitting vectors* $\{u_j^i\}, i = 1, 2, \ldots, M$, *associated with a set of suboperators* A_i, *where* (6.1) *is satified. Suppose* P_i *is the matrix Padé approximant for submatrix* A_i, *and* $u_{j+1}^i = P_i u_j^i$. *if for any integers* i *and* k, *where* $1 \leq i, k \leq M$, $P_k u_j^i = u_j^i$ *if* $k \neq i$, *the set of* $\{u_j^i\}$ *is an* **Independent set of partial solutions.**

Now if $\{u_j^i\}$ is an independent set of partial solutions, by the BSPTS prescription (5.11),

$$\begin{aligned} & u_{j+1} = P_n^{n,t_j}(hA)u_j, \\ \Rightarrow \quad & u_{j+1} = P_n^{n,t_j}(hA)(u_j^1 + u_j^2 + \cdots + u_j^M) \\ \Rightarrow \quad & u_{j+1} = P_n^{n,t_j}(hA)u_j^1 + P_n^{n,t_j}(hA)u_j^2 + \cdots + P_n^{n,t_j}(hA)u_j^M) \\ \Rightarrow \quad & u_{j+1} = P_n^{n,t_j}(hA_1)u_j^1 + P_n^{n,t_j}(hA_2)u_j^2 + \cdots + P_n^{n,t_j}(hA_M)u_j^M, \end{aligned}$$

where $P_n^{n,t_j}(hA_i)$ denotes the items only depending on A_i in $P_n^{n,t_j}(hA)$. We should note that, each $P_n^{n,t_j}(hA_i)$, in fact is a product of certain terms of $P^n(hA_i)$ for certain order n, i.e., $P_n^{n,t_j}(hA_i) = (P^n(hA_i))^m$ for certain integers m and n. Therefore, if $|P^n(hA_i)| \leq 1$, then $|(P^n(hA_i))^m| \leq 1$, thus $|P_n^{n,t_j}(hA_i)| \leq 1$.

Based on the analysis above, with the nature of BSPTS schemes, we can obtain a general stability result as follows:

Theorem 6. *Let A be a real $k \times k$ matrix, with its submatrices $\{A_i\}$, i=1, 2, ..., $M, M < k$, such that $A = \sum_{i=1}^{M} A_i$, and there exists an independent set of partial soluions $\{u_j^i\}$, associated with $\{A_i\}$. Then the BSPTS$[N, N]$ scheme for the system of ODEs $\dot{u} = Au$ is stable for all step size $h > 0$, if all eigenvalues of each submatrix are real and nonpositive.*

Proof. Let i be any integer $\in [1, M]$ and $\lambda_{i,p}$ denote the p-th eigenvalue of A_i. Suppose that for each submatrix A_i with a step size $h > 0$, by BSPTS$[N, N]$ scheme, $u_{j+1}^i = P_i u_j^i$. Since $\lambda_{i,p} \leq 0$ for all possible p and i, then $|P^n(hA_i)| \leq 1$, therefore,

$$|P_i| = |(P^n(hA_i))^m| \leq 1 \Rightarrow |P_n^{n,t_j}(hA_i)| \leq 1$$

let

$$u_j = u_j^1 + u_j^2 + \cdots + u_j^M, \text{ where } u_j^i \subset u^i,$$

and

$$u_{j+1} = P_1 u_j^1 + P_2 u_j^2 + \cdots + P_M u_j^M.$$

By Theroem 2, $|P_i| \leq 1$ for all i along with the independence of $\{u_i\}$, which implies that BSPTS$[N, N]$ schemes are stable for all step sizes $h > 0$. □

Most numerical schemes are extraordinarily general, in that they are designed for use with arbitrary matrices. In practice, however, most applications are far more specialized, and each applications may require different implementation from others. For example, standard discretization of partial differential equations typically lead to large and sparse matrices. A sparse matrix is defined [25], somewhat vaguely, as a matrix which has very few nonzero elements. The sparse matrices can be essentially distinguished into two broad types: *structured* and *unstructured*. A structured matrix is a block matrix, i.e., a matrix of dense submatrices (blocks) of the same size, whose blocks form a regular pattern, typically along a small number of block-diagonals. In contrast, a matrix with irregularly located entries is said to be unstructured. Finite difference matrices on rectangular grids are typical example of structured matrices. Even for those unstructured matrices, by the Jordan decomposition [18] techniques, a square matrix can always be converted to the block diagonalized Jordan form where our results can apply.

Based on the nature of the operator, a certain splitting can be applied to the operator to ensure the stability of overall schemes. However, the conversion of stability property of MPTS to BSPTS can be too complicated or even impossible, except those operators which can be split into several independent suboperators where the Theorem 6 can be refered. In what follows, we present a simple case to illustrate the application of Theorem 6 to more specified cases.

Diagonal matrices operators

A diagonal matrix A may be the simplest operator, which may, for instance, arise from some i.i.d. random phases problems. For this case we make a conclusion

Theorem 7. *Let A be a real $k \times k$ matrix with diagonal entries a_i, i=1, 2, ..., k. If $a_i < 0$ $\forall i$, then BSPTS$[N, N]$ scheme for the system of ODEs $\dot{u} = Au$ is stable for all step sizes $h > 0$, and BSPTS$[N, N]$ numerical solution u_j for this system tends to zero as $j \to \infty$.*

Proof. Let

$$A = \begin{bmatrix} a_1 & 0 & 0 & \dots & 0 \\ 0 & a_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{k-1} & 0 \\ 0 & \dots & 0 & 0 & a_k \end{bmatrix}, \quad a_j \leq 0, \quad j = 1, 2, \dots, k.$$

By BSPTS$[N, N]$, suppose A is split into M submatrices, such that each submatrix, in its original splitting form

$$A_i = \begin{bmatrix} 0 & & & & & & & \\ & \ddots & & & & & & \\ & & 0 & & & & O & \\ & & & \tilde{a}_{i,j} & & & & \\ & & & & \ddots & & & \\ & & & & & \tilde{a}_{i,j+k-M+1} & & \\ & & O & & & & 0 & \\ & & & & & & & \ddots \\ & & & & & & & & 0 \end{bmatrix},$$

where $A = \sum_{i=1}^{M} A_i$, $\tilde{a}_{i,j} = a_j$. Since all eigenvalues of A_i are nonpositive $\forall i$. Therefore the results we claimed just follows after Theorem 6. □

Note that, although in this thesis we do not give the theoretic proof for stability on other type of operators, numerical examples do suggest that BSPTS schemes possess of strong stability property on a wide range of operators arising from various popular problems, for instance, one way wave equations of hyperbolic PDEs and diffusion equation of parabolic PDEs. The theoretic proof to verify what those numerical examples suggest is in progress.

Chapter 7

Numerical Experiments

7.1 Numerical experiments: Ordinary Differential Equations

We begin the numerical tests with a sparse system of ODEs which easily makes explicit schemes unstable. By testing this simple problem, we can also verify the order of accuracy of the methods. Consider a 5×5 system of linear ODEs

$$\dot{u} = Au, \text{ with a given initial value } u(0) = u_0,$$

where

$$A = \begin{bmatrix} -180 & -1 & 0 & 0 & 0 \\ 5 & -1 & -2 & 0 & 0 \\ 0 & -1 & -20 & -1 & 0 \\ 0 & 0 & -3 & -4 & 5 \\ 0 & 0 & 0 & 3 & -10 \end{bmatrix}, \quad u_0 = \begin{bmatrix} .01 \\ .1 \\ 2 \\ 10 \\ 100 \end{bmatrix}.$$

Clearly, the eigenvalues are disparate, which render this an ill-conditioned

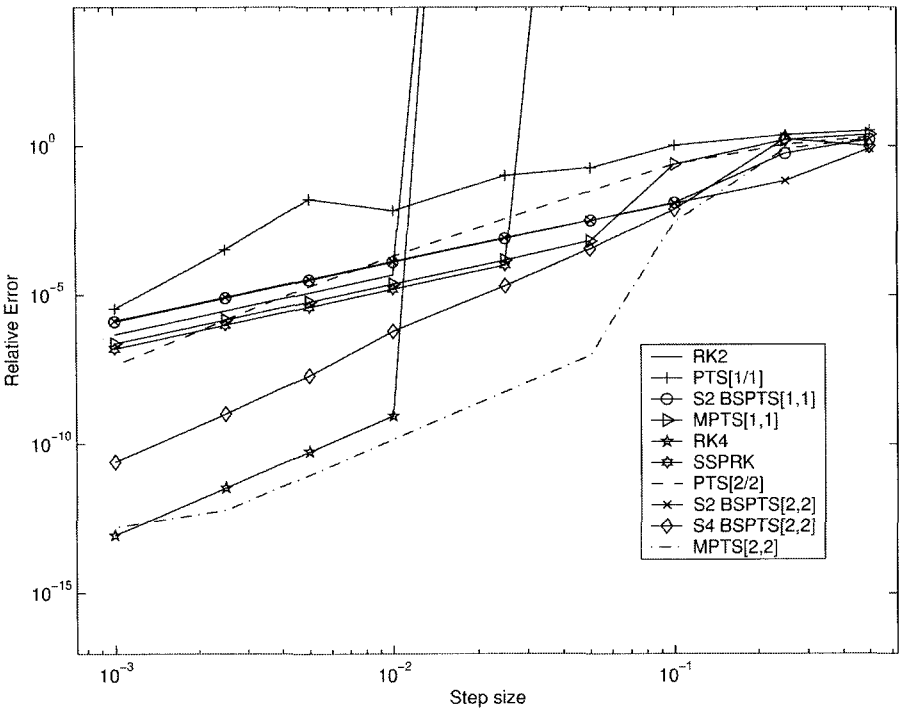

Figure 7.1: *Stabilities of BSPTS schemes with other methods on a sparse system of ODEs*

problem for explicit schemes. In this example, we compare BSPTS[1,1], S_2 BSPTS[2,2] and S_4 BSPTS[2,2] schemes with PTS[1,1] and PTS[2,2], and also with RK2, RK4, and SSPRK.

To quantify the accuracy of the computed solutions, we compare the step-sizes and relative errors. Figure 7.1 has firstly confirmed the expected order of accuracy for each method. It also shows that, for small time-steps BSPTS schemes perform as well as the PTS and RK schemes, and that they continue to provide stable and accurate results well beyond the RK stabilities. We also note that, BSPTS schemes can provide even higher accuracy than PTS schemes by $10^2 \sim 10^4$ times in this case.

7.2 Numerical experiments: Partial Differential Equations

Now we study the numerical behavior of our schemes for a few PDE problems designed to capture solution features that pose particular difficulties to numerical methods. Experiments for the PTS and MPTS schemes are included in every test example, while the classical 2nd and 4th order explicit Runge-Kutta methods are also included because those methods are commonly used but not strongly stable for certain cases. In addition, the Strong-Stability-Preserving Runge-Kutta (SSPRK) [27] method is included as well, which gives us an even more visible basis for comparison given its strong stability features.

To investigate the behavior of our BSPTS schemes, we consider three types of PDEs, which are the one way wave equation (a linear hyperbolic PDE), the one dimensional diffusion equation (a linear parabolic PDE), and the KdV equation (a nonlinear hyperbolic PDE). In all test problems, we focus on the behavior of the numerical schemes for interior regions rather than boundaries and impose periodic boundary conditions on certain domains. It is known that sometimes a conventional (and intuitive) treatment of the boundary data (especially in the case of inflow boundary conditions) within the stages of an explicit method, for example a Runge-Kutta method, can lead to a deterioration in the overall accuracy of the integration.

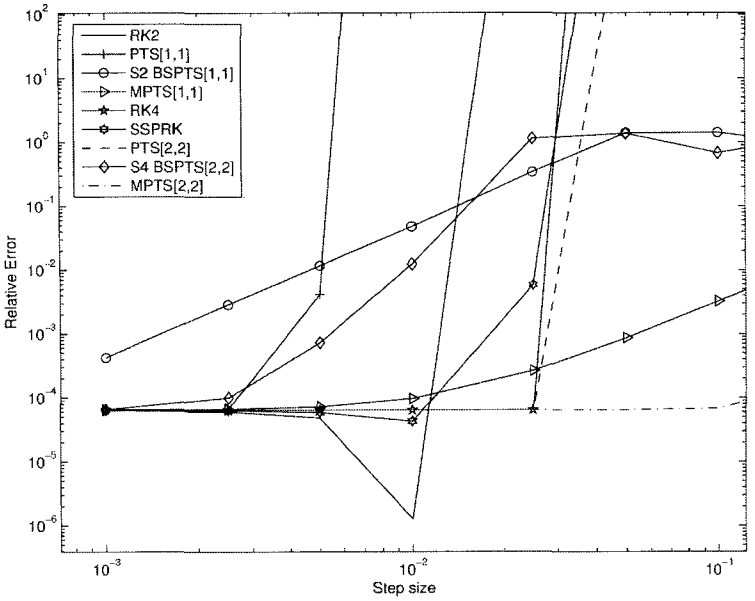

Figure 7.2: *Stabilities of BSPTS schemes comparing with various methods on one way wave equation* $u_t + u_x = 0$

7.2.1 Hyperbolic PDEs

One way wave equation

In this case, the initial condition

$$u(x, 0) = e^{-x^2}$$

is evolved to time $t = 1$ according to the linear advection equation (2.16), also called one way wave equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

using a constant grid spacing of $\Delta x = 0.01$. It is clear that the exact solution is $u(x, t) = e^{-(x-t)^2}$ on the domain [-10, 10]. A plot of the relative errors is given in Figure 7.2.

In this example, all BSPTS schemes can give improved stability over the PTS

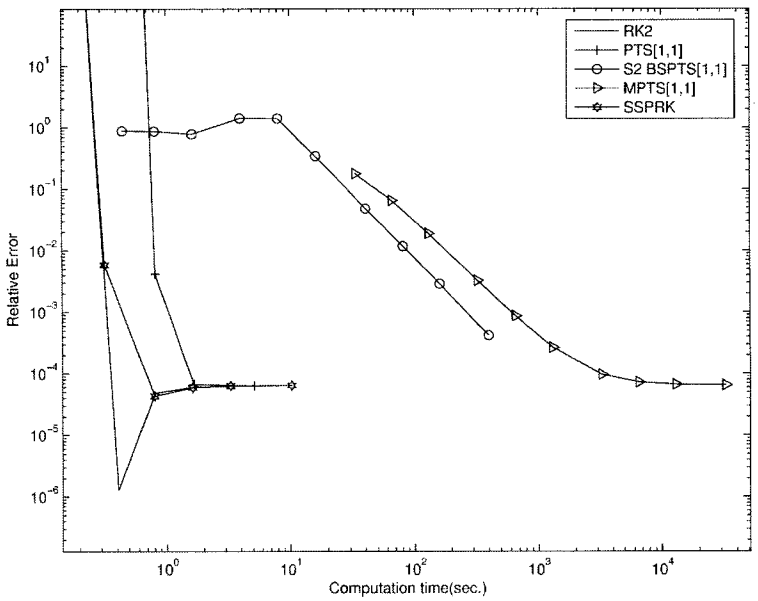

Figure 7.3: *The computation efficiencies of BSPTS schemes vs. various methods on one way wave equation* $u_t + u_x = 0$

schemes by their unconditional stability property on this type of problem. On the other hand, obviously the MPTS scheme provides the best stability and accuracy among all methods. However, if we look at the Figure 7.3, we just find that S_2 BSPTS[1,1] shows its computational efficiency over MPTS[1,1].

KdV equation

The next test example is the Korteweg de Vries equation (KdV) equation, which in canonical form, is given by

$$u_t + u_{xxx} + 6uu_x = 0.$$

Clearly the KdV equation is a nonlinear PDE problem, whose solution is no longer given formally by

$$u = e^{tA}u_0,$$

since such matrix A varies at each step. However, at each step we still take this changing operator, split and apply BSPTS in order to find its numerical solution. The reason we can do so is that, on a step by step basis, we have effectively "frozen" the matrix at time t_n by taking appropriate step size dt and solving the resulting (constant coefficient) linear system. However when such dt is too large so that the approximation of frozen A is no longer valid, the accuracy of the solution will be lost.

The KdV equation with the single solition initial condition $u_0 = u(x,0) = 2\text{sech}^2(x)$, for which the exact solution is known as $u(x,t) = 2\text{sech}^2(x-4t)$ [1] on the truncated domain $x \in$ [-10,10] is evolved. In Figure 7.4 we present

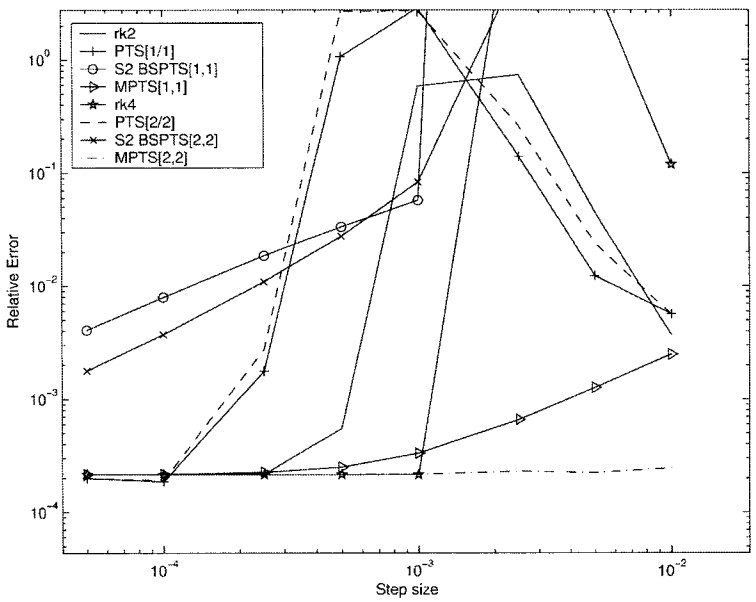

Figure 7.4: *BSPTS schemes vs. various methods on KdV equation*

the step-sizes vs. the relative errors arising from applications of some 2nd order methods, i.e., the S_2 BSPTS[1,1] with RK2 and SSPRK to the solution of the KdV equation up to time $t = \frac{1}{4}$, and the splitting factor s is randomly chosen from $[0,1]$.

As we can see from this plot, although all methods fail to stay stable when the step size becomes large, S_2 BSPTS[1,1] which loses its stability when $dt > 0.001$, has shown the advantage over PTS[1,1] which goes to unstable when $dt > 0.05$.

7.2.2 Parabolic PDEs

Diffusion equation

The parabolic PDE problems have restricted the usage of general block splitting methods because Sheng [26] has showed that higher order (order greater than two) operator splitting methods must contain operators which integrate backwards in time. In the same study, Sheng also showed that a certain class of split operators containing sums of product terms, some terms having backward time evolution, were unstable in integrating a split heat equation. Therefore, the conclusion for this class of split operators was that they were unstable for higher order methods.

In section 5.2, we have confirmed that in the diffusion equation cases, it is true that higher order splitting methods all must have negative time evolution operators. Nevertheless, they are not unstable, at least not all unstable when we approximate the split blocks via their Padé approximants by providing an appropriate splitting factor s. In fact, there is one and exactly one splitting factor $s = \frac{1}{2}$ which makes higher order BSPTS schemes (4th order in our test case) stable for diffusion equations. A wide range of splitting factors have been tested, we indeed see that any s other than $\frac{1}{2}$ can make this scheme unsuccessful.

We have tested the diffusion equation

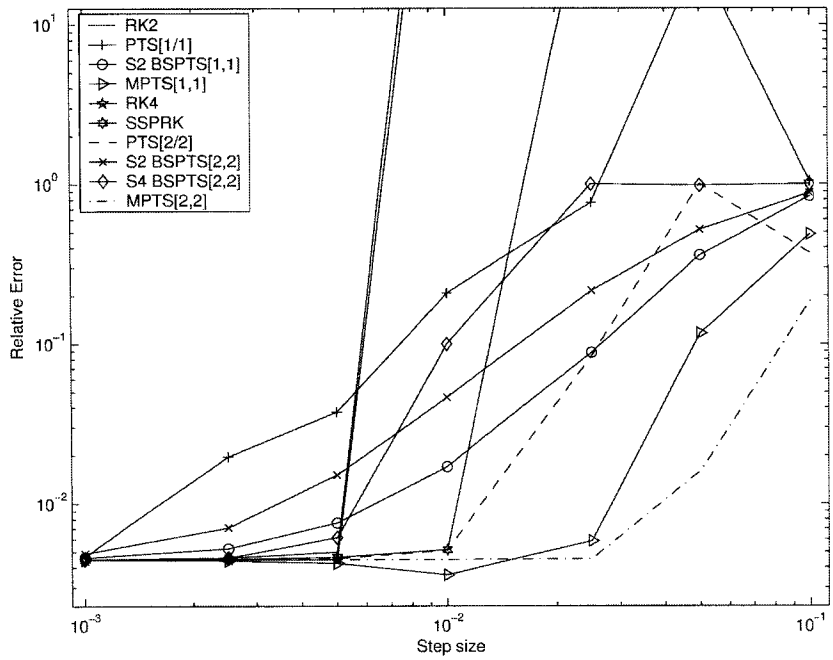

Figure 7.5: *BSPTS schemes vs. various methods on heat equation* $u_t = u_{xx}$ *with* $s = \frac{1}{2}$

$$u_t = u_{xx}, \quad t > 0$$

on the truncated domain of $x \in$ [-10,10], with the initial condition and the solution are given by [22]

$$u(x,t) = \frac{1}{\sqrt{4\pi(\rho+t)}} e^{\frac{-x^2}{4(\rho+t)}}$$

where ρ=0.1.

A plot of relative errors for evolving this problem by BSPTS schemes with the splitting factor $s = \frac{1}{2}$ is shown on Figure 7.5. From this plot, we can clearly see that BSPTS schemes are presenting their favorable unconditional stability given an appropriate splitting factor. On the other hand, we have also tested this problem with values of s other than $\frac{1}{2}$ and the relative errors plots are given on Figure 7.6. The fourth order scheme S_4 BSPTS[2,2] is unstable for all other values of s as the relative errors blow up when the step size $dt \geq \frac{1}{2}$. The

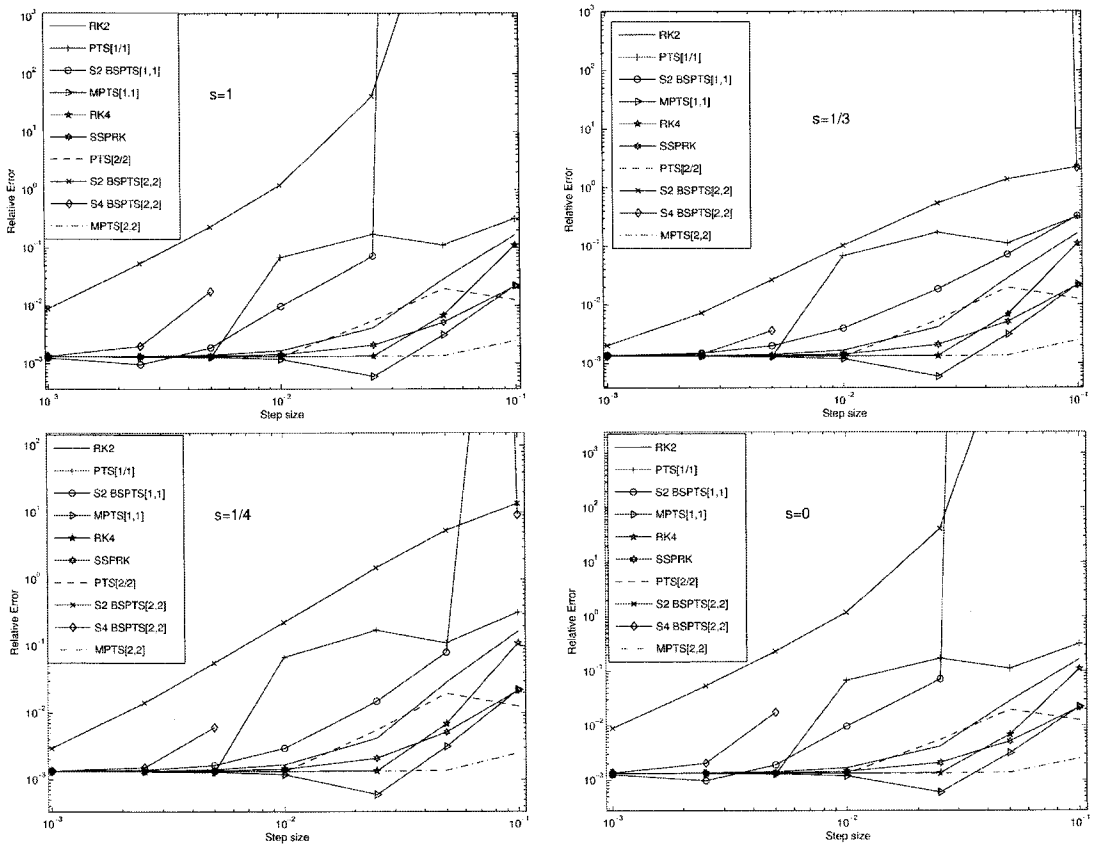

Figure 7.6: *Convergence of BSPTS schemes on heat equation* $u_t = u_{xx}$ *with a range of splitting factors*

second order schemes S_2 BSPTS[2,2] and S_2 BSPTS[1,1] stay unconditionally stable when the value of s is close enough to $\frac{1}{2}$ (for instance $s = \frac{1}{3}$ in our test), otherwise they cannot hold the stability for larger step sizes neither.

7.3 Alternative splitting methods

In this context so far we have only discussed a few splitting techniques of this big family of time-stepping schemes, which may be named ***diagonal block splitting*** based on certain operator matrices with a band of diagonal blocks. While such type of matrices may cover a wide range of operators, the BSPTS schemes

could also be applicable to evaluate problems with more general operators derived from more general splitting techniques within the range of satisfying the essential condition (5.1). Here we present a simple example where an alternative splitting method is applied. In this example, the one way wave equation is evolving by a more specific block splitting Padé approximation, namely ***Row Block-splitting PTS (RBPTS)***, which can enlarge the step size for the stability to PTS for 3-4 times while cost only 50% more computational time per step.

The operator A arises from equation (2.16). Suppose A is dimension of $n \times n$,

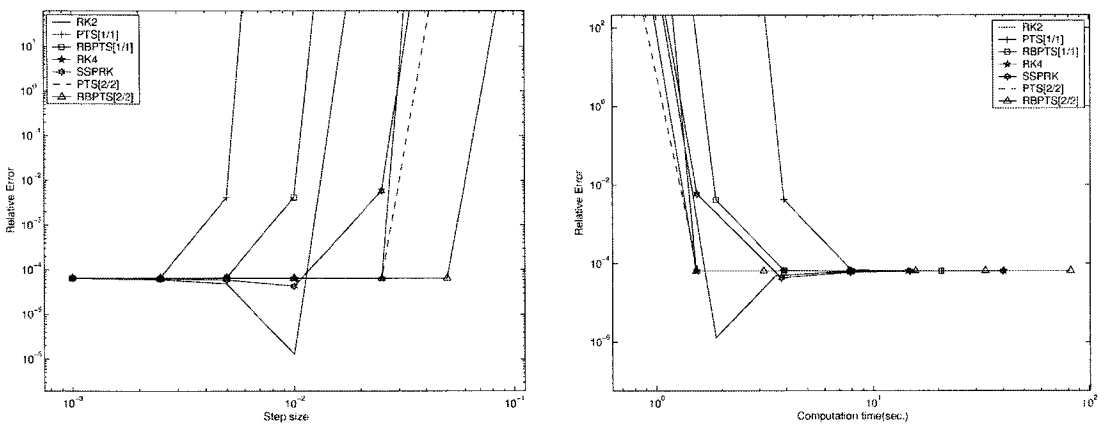

Figure 7.7: *Relative performance of RBPTS schemes for one way wave equation*

by means of RBPTS, A is split into n row blocks, such that there is no overlap between blocks. If a $[N/N]$ Padé approximation defined by (2.7) exists for each row block m at time t_j with step size h, that is, if $[N/N]^{t_j}(h)$ exists, combinating with the general block splitting algorithm,

$$\begin{aligned} u_{j+1} &= [N/N]^{t_j}(\tfrac{h}{2})u_j[N/N]^{t_j}(\tfrac{h}{2})u_j = R_2(h)u_j && \text{2nd order accuracy} \\ u_{j+1} &= R_2(\omega h)R_2((1-2\omega)h)R_2(\omega h)u_j && \text{4th order accuracy} \end{aligned} \tag{7.1}$$

...

A relative numerical experiment has been done, and the convergency and efficiency plot are given in Figure 7.7. We clearly see, RBPTS[1,1] and RBPTS[2,2] improve the stability by around 3-4 times to PTS[1,1] and PTS[2,2] respectively. Especially, for certain order of accuracy RBPTS[1,1] performs 4 times more efficient than PTS[1,1], which we don't achieve by the diagonal block splitting.

However, in this case RBPTS is not unconditionally stable, and RBPTS[1,1] is still even less stable than RK2. As we expect, higher stability RBPTS schemes may be obtained from the technique, namely ***multi-componentwise Padé approximation***, which Padé approximates the Taylor expansions of the unknowns on the multi-component manner [6]. The investigation on this topic is in progress.

Chapter 8

Summary and Future Work

In summary, we have presented a new class of time-stepping schemes of block splitting of exponential operator via their Padé approximants, named BSPTS schemes, which encompass the highly stable but computationally expensive MPTS[N,N] methods, and the very efficient but less stable PTS[N,N] schemes which perform the algebraic computations componentwise. We have shown that, for a wide range of operators arising from various systems of DEs, BSPTS holds the unconditional stability. We have also performed a comparison of the new schemes with Runge-Kutta methods commonly used in practice on a nonlinear PDE problem (KdV equation). Our new schemes compare favorably in terms of stability. We find that, the BSPTS schemes not only hold the unconditional stability property in many popular problems, for instance one way wave equations, but also give rise to higher computational efficiency than the implicit MPTS schemes on a per time-step basis.

The improvements are the greatest for higher order BSPTS schemes applied to parabolic PDE problems where general high order block splitting methods fail.

We find that there is one, and exactly one value of splitting parameter which can make higher order splitting methods stable in evolving diffusion equations.

We should note that while examples presented here are more for the purpose of illustration of the methods, in practice, problems where the systems are structured but stiff, are most suitable to this approach.

However, as an 'intermediate' approach between stable schemes and efficient methods, in this thesis we actually study the BSPTS schemes from the viewpoint of stability. In order to obtain high stability, we simultaneously pay a price on losing efficiency per step. For example, we can read that from Figure 7.3, BSPTS schemes are far less efficient than explicit methods RK and PTS ($10^2 \sim 10^4$ times slower) for certain order of accuracies. Incorporating efficient linear algebra methods, for example, iterative mathods, and multi-step methods to speed up the computation per step would make this scheme more practicable.

On the other hand, this also motivates us to alternately explore BSPTS methods from the viewpoint of efficiency. As the last example indicates, based on the structure of certain operator, alterative splitting methods, including *row block splitting*, *triagonal block splitting*, *unequal block splitting*, etc., should be considered to achieve an better performance, in terms of accuracy, stability and efficiency. Further investigation about the algebra concerned in deriving high performing methods and their properties is left for future work.

Bibliography

[1] M. J. Ablowitz and P. A. Clarkson. *Solition, nonlinear evolution equations and inverse scattering*, London mathematical Society Lecture Note Series, Vol.149 Cambridge Univ. Press, (1991).

[2] I. D. Abrahams, *The application of Padé approximants to Wiener-Hopf factorization*, IMA J. Appl. Math. Vol.65 (2000) *pp.*257-281.

[3] G. P. Agrawal, *Nonlinear Fiber Optics*, Second Edition. Academic Press. (1995).

[4] D. Amundsen and O. Bruno. *Time stepping via one-dimensional Padé approximation*, J. Sci. Comp. To appear (2006).

[5] J. H. Argyris, P.C. Dunne and T. Angelopoulos, *Dynamic response by large step integration, Earthquake Engrg*, Structural Dynam. Vol.2 (1973) *pp.*185-203.

[6] G. A. Baker, Jr. and P. Graves-Morris. *Padé Approximants Part1: Basic theory*, Encyclopedia of Mathematics and Its Applications, Vol.13 Addison-Wesley, (1981).

[7] A. D. Bandrauk and H. Shen, *High-order split-step exponential methods for solving coupled nonlinear Schrödinger equations*, J. Phys. A: Math. Gen. 27 (1994) *pp.*7147-7155.

[8] G. Birkhof and R. S. Varga. *Discretization errors for well-set Cauchy problems I*, J. Math. Phys. Vol.44 (1965) *pp.*1-23.

[9] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Second Edition. Dover. (2001).

[10] C. Brezinski and J. van Iseghem, *Padé approximations*, in *Handbook of Numerical Analysis*, vol. III, P.G. Ciarlet and J.L. Lions eds., North-Holland, Amsterdam (1994), *pp.*47-222.

[11] C. Brezinski and J. van Iseghem, *A taste of Padé approximation*, Acta Numerica 1995, A Iserle ed., Cambridge UniversityPress, Cambridge (1995), *pp.*53-103.

[12] J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, John Wiley & Sons. (1987)

[13] A. Draux and B. Moalla. *Rectangular matrix Padé approximants and square matrix orthogonal polynomials*, Numerical Algorithms. Vol.14 (1997) *pp.*321-341.

[14] D. Goldman and T. J. Kaper, *Nth-order operator splitting schemes and nonreversible systems*, SIAM J. Numer. Anal. Vol.33, No.1, (1996) *pp.*349-367.

[15] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Second Edition, Springer. (1996)

[16] M. Hochbruck, C. Lubich, and H. Selhofer, *Exponential integrators for large systems of differential equations*, SIAM J. Sci. Comput., Vol.19, (1978) *pp.*1552-1574.

[17] H. Holden, K. H. Karlsen and K. Lie, *Operator splitting methods for degerate convection-diffusion equations II: numerical examples with emphasis on reservoir simulation and sedimentation*, Comp. Geosci. Vol.4 (2000) *pp.*287-322

[18] http://en.wikipedia.org/wiki/Jordan_normal_form

[19] F. John, *Partial Differential Equations*, Fourth Edition. Springer-Verlag (1982).

[20] R. J. LeVeque and J. Oliger, *Numerical methods based on Additive Splittings for Hyperbolic Partial Differential Equations*, Math. Comp. Vol.40, No.162 (1983) *pp.*469-497

[21] R. I. McLachlan, R. Quispel, *Splitting methods*, Acta Numer. 11 (2002) *pp.*341C434.

[22] T. Meis and U. Marcowitz, *Numerical Solution of Partial Differential Equations*, Springer-Verlag (1981).

[23] C. Moler and C. van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Review, Vol.20, No.4 (1978) *pp.*801-836.

[24] W. H. Press et. al. *Numerical Recipes in Fortran 77: The art of scientific computing*, cambridge University Press, (1996).

[25] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Inc (1996).

[26] Q. Sheng, *Solving linear partial differential equations by exponential splitting*, IMA J. Numer. Anal. 9 (1989) *pp.*199-212

[27] A. J. Spiteri and S. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM Journal on Numerical Analysis, Vol.40, No.2 (2002) *pp.*469-491

[28] M. Suzuki, *Convergence of general decompositions of exponential operators*, Commun. Math. Phys. 163, (1994) *pp.*491-508

[29] M. Suzuki, *General theory of fractal path integrals with applications to many-body theories and statistical physics*, J. Math. Phys., Vol.32, No.2, (1991) *pp.*400-407

[30] M. Suzuki, *Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations*, Phys. Lett. A 146, (1990) *pp.*319-323.

[31] M. Suzuki, J. Math. Phys. 26, (1985) *pp.*601.

[32] M. Suzuki, Phys. Lett. A 113,(1985) *pp.*299.

[33] A. Wragg and C. Davies, *Computation of exponential of a matrix II, Practical considerations*, Ibid. 15 (1975) *pp.*273-178.

[34] G. Xu and A. Bultheel, *Matrix Padé approximation: definitions and properties*, Linear algebra and its applications, V. 137/138 (1990) *pp.*67-136

[35] V. Zakian, *Rational approximants to the matrix exponential*, Electron. Lett. 6, (1970) *pp.*814-815.

[36] D. Zwillinger, *Handbook of Differential Equations*, Second Edition. Academic Press, (1992). *pp.*392.