Journal of Computational and Applied Mathematics 233 (2009) 1056–1062

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Numerical smoothing of Runge–Kutta schemes

Tong Sun *

Department of Mathematical and Statistics, Bowling Green State University, Bowling Green, OH 43537, USA

ARTICLE INFO

Article history:
Received 23 January 2009
Received in revised form 2 July 2009

ABSTRACT

First we give an intuitive explanation of the general idea of Sun (2005) [1]: consistency and numerical smoothing implies convergence and, in addition, enables error estimates. Then, we briefly discuss some of the advantages of numerical smoothing over numerical stability in error analysis. The main aim of this paper is to introduce a smoothing function and use it to investigate the smoothing properties of some familiar schemes.

1. Numerical stability vs. numerical smoothing

The problems that we consider can be represented abstractedly, using Banach space terminology, as in the text of Richtmyer and Morton [2]. We wish to find a solution of the evolution equation

$$u_t = F(u), \quad t > 0 \tag{1.1}$$

with the initial condition

$$u(0) = u_0, \tag{1.2}$$

where F is a nonlinear operator from the Banach space $\mathcal{B}$ containing u to a proper range space. For more details of the abstract problem, one can go to Chapter 3 of the reference [2], where a linear problem is presented. Here we will not specify the details as regards the well-posedness of our abstract problem. However, one can find more details in concrete cases from [1,3,4]. We choose to consider an autonomous problem in order to avoid tedious notation. For PDEs, a reader can find how to deal with spatial discretization error in the context of numerical smoothing from [1,4]. Here we focus on the discretization of time.

Numerical stability concerns how a numerical scheme propagates error. Fig. 1 gives an explanation. Let $u(t)$ stand for a solution of the evolution equation (1.1) and $u_N(t)$ stand for a numerical solution obtained by using a numerical scheme. The total error at t_{n+1} is split into local error and propagation error by $\breve{u}_N(t_{n+1})$,

$$u(t_{n+1}) - u_N(t_{n+1}) = u(t_{n+1}) - \breve{u}_N(t_{n+1}) + \breve{u}_N(t_{n+1}) - u_N(t_{n+1}). \tag{1.3}$$

Here $\breve{u}_N(t_{n+1})$ is the non-computable numerical solution which would be obtained by using the same scheme, in the step $[t_n, t_{n+1}]$, with initial value $u(t_n)$.

In most textbooks, a typical way of defining local error is to put the real solution $u(t)$ in a scheme to find the residual, which is equivalent to looking at $u(t_{n+1}) - \breve{u}_N(t_{n+1})$. The reason for defining $u(t_{n+1}) - \breve{u}_N(t_{n+1})$ to be the local error is quite obvious: the smoothness of $u(t)$ can be used in this way.

As a consequence of this local error definition, one has to face the second part of the split error, namely, $\breve{u}_N(t_{n+1}) - u_N(t_{n+1})$, which is clearly the numerical propagation of the error $u(t_n) - u_N(t_n)$. Usually, one tries to prove, in an appropriate norm,

$$\|\breve{u}_N(t_{n+1}) - u_N(t_{n+1})\| \leq |1 + \beta\tau_n| \|u(t_n) - u_N(t_n)\|, \tag{1.4}$$

* Tel.: +1 419 372 7460; fax: +1 419 372 6092.
E-mail address: tsun@bgsu.edu.

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.08.118

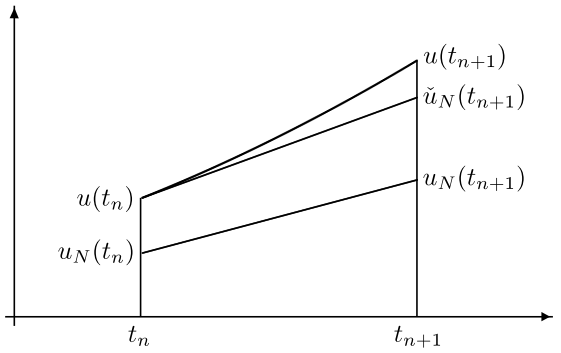

Fig. 1. Numerical stability.

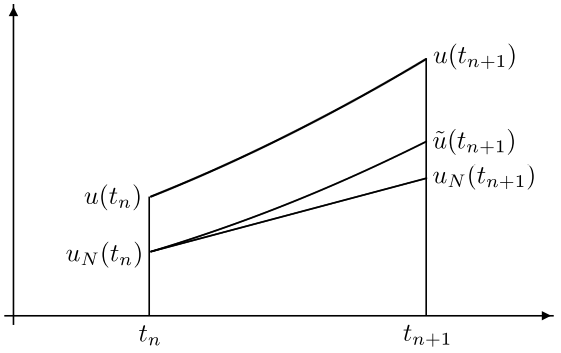

Fig. 2. Numerical smoothing.

where $\tau_n = t_{n+1} - t_n$ is the step size, β is a constant depending on the scheme, but not on τ_n. It would be ideal to have $|1+\beta\tau_n| \le e^{-\gamma\tau_n} < 1$, with a positive γ, so that one could prove a uniform convergence of the numerical solution. If $\beta = 0$, there will be a simple accumulation of local error into global error. Unfortunately, one often ends up getting a positive β. Both the nonlinearity of the evolution equation and the complexity of the numerical scheme will contribute to making β larger. In the global error estimate, one will consequently see a factor $e^{\beta T}$ if the numerical solution is to be computed in $[0, T]$. In many cases, proving (1.4) is simply too hard due to nonlinearity, scheme complexity, and other technical difficulties.

For linear evolution equations, there is the well-known Lax–Richtmyer Theorem [6]. *Assuming that a scheme is consistent with the equation, numerical stability is necessary and sufficient for convergence.* Because it is often too difficult to prove numerical stability, many problems are solved without proper error estimates. This leaves a huge gap between error analysis theory and scientific computation practice.

In Fig. 2, the error is split in a different way. $u(t)$ and $u_N(t)$ are defined as before. For the entire time domain, $\tilde{u}(t)$ is defined piecewise. In each time step (t_n, t_{n+1}), $\tilde{u}(t)$ is the solution of the evolution equation (1.1) with initial value $u_N(t_n)$. Here the total error at t_{n+1} is split as

$$u(t_{n+1}) - u_N(t_{n+1}) = u(t_{n+1}) - \tilde{u}(t_{n+1}) + \tilde{u}(t_{n+1}) - u_N(t_{n+1}). \tag{1.5}$$

The first difference on the right hand side is the propagation of $u(t_n) - u_N(t_n)$ by the evolution equation. The second difference is the actual local error, which is the error of $u_N(t_{n+1})$ while approximating $\tilde{u}(t_{n+1})$. For readers who are not yet familiar with the idea of numerical smoothing for error analysis, the following explanation may be important and helpful.

In the error splitting of Fig. 2, there is no position for numerical error propagation. The error propagation by the evolution equation is exactly how error should be propagated. However, there is a new issue, which leads to the concept of numerical smoothing. Let's address this issue in three logically connected steps.

1. The smoothness of $\tilde{u}(t)$ needs to be established in (t_n, t_{n+1}), so that the local error $\tilde{u}(t_{n+1}) - u_N(t_{n+1})$ can be estimated.
2. Since $\tilde{u}(t)$ is a solution of a well-posed evolution equation, its smoothness is determined by its initial value $u_N(t_n)$.
3. $u_N(t_n)$ is the result of the numerical computation in the past before time t_n; the scheme or schemes that we have used so far should have had some kind of numerical smoothing effects, such that $\tilde{u}(t)$ has smoothness.

Once we have established the smoothness of $\tilde{u}(t)$, we can obtain a local error estimate for $\tilde{u}(t_{n+1}) - u_N(t_{n+1})$ by using the consistency of the schemes.

What do we mean by a "**smoothing effect**" of numerical schemes? For ODE systems, the stiff components in the numerical solution $u_N(t_n)$ should have been reduced to near zero, so that the fast changing components in $\tilde{u}(t)$ are small enough. For parabolic problems, the high frequency components in space should have been dissipated. Since the high

frequency components in space are also the fast changing components in time, a dissipative scheme leaves $\tilde{u}(t)$ smooth. For hyperbolic problems, the story is more complicated. The word smoothing seems to be not appropriate, because the formation and evolution of shock waves should not be damped out. Nevertheless, some kind of treatment is needed to avoid numerical dispersion and anti-diffusion. The discussion in this paragraph is only intended to be intuitive. Rigorous analysis has to be based on the definition of a smoothing indicator for each concrete case. See [1,3,4] for examples.

The above approach of error propagation analysis has been used in [1,3,4] for long time error analysis. In these papers, some examples of nonlinear problems and complex schemes are presented. There are also details about how to apply the idea to parabolic equations with space discretization. Here we summarize the idea in the following simple theorem, to indicate that numerical stability can be replaced by numerical smoothing.

Theorem 1.1. *Consistency and numerical smoothing ⇒ convergence.*

In many textbooks, one can find explanations as regards the concept of numerical stability. Typically, the understanding is that stable schemes do not amplify small local truncation error too badly, while unstable schemes amplify small error too much and ruin the numerical solutions. When we work with numerical smoothing instead of numerical stability, we have a totally different understanding. We believe that error propagation is only done by the evolution equation. If we do not have enough numerical smoothing, we produce a large local error at each step. If we use an anti-smoothing scheme, we produce larger and larger local errors step after step. If we use a scheme with some numerical smoothing properties and adaptively choose time step sizes, we can control both local and global error.

2. Advantages of smoothing analysis over stability

There are several advantages of using numerical smoothing over numerical stability in error analysis. The fundamental difference between numerical smoothing and numerical stability is the following. Numerical stability concerns the relation between two numerical solutions; one of them is always not computable. Numerical smoothing only concerns one numerical solution, namely the computed one. The impact of this difference reaches far and wide.

1. Since only the computed numerical solution is involved, loss of superposition in a nonlinear problem will not stop the computation of a smoothing indicator. Hence numerical smoothing analysis works better for nonlinear problems.
2. No matter what numerical schemes are used for the time discretization, the smoothing indicator at each t_n is computed in the same way. Hence an error estimate can be obtained regardless of scheme complications.
3. Error propagation is estimated by using evolution equations instead of schemes. All the potential troubles of numerical error propagation analysis are avoided. After all, error propagation by evolution equation is exactly how error should be propagated. Without complication of a numerical scheme, one can apply any contraction properties of the ODE and PDE solutions, including contraction of a solution, a limit cycle, a chaotic attractor, or any other attractive objects useful for error analysis. This is actually what makes long time error estimation possible [3,4].
4. Adaptive time stepping is natural, since the regularities of the solutions $\tilde{u}(t)$ are revealed by the smoothing indicator in each step $[t_n, t_{n+1}]$ [1].
5. The additional cost of computing the smoothing indicator is almost negligible. As shown in [1,3,4], computing the smoothing indicator involves a few actions with a stiffness matrix and a mass matrix inversion. The stiffness matrices are computed anyway, while the mass matrix inversion can be avoided by mass lumping.

3. Smoothing function

Although we can rely on a smoothing indicator for local error estimates and adaptive error control, we still like to know what kinds of schemes are more likely to have a smoothing effect, and what kinds of schemes do not. As a beginning step, we study the Runge–Kutta family of schemes. In this paper, we will only give a coarse analysis in general. Except for a few simplest cases, such as the Euler and Crank–Nicolson schemes, more detailed studies are to be done in the future in order to get sharp smoothing effect estimates.

The model equation

$$\dot{y} = \lambda y \tag{3.1}$$

has been used in analyzing the stability of time stepping schemes. We also use this model problem to analyze numerical smoothing properties. In this paper, we assume that λ is in the set $\mathcal{S} = \mathcal{S}_M \cup \mathcal{S}_\alpha$, where $\mathcal{S}_M = \{\lambda \in \mathbb{C} : |\lambda| \le M < \infty\}$, $\mathcal{S}_\alpha = \{\lambda \in \mathbb{C} : |\mathrm{Im}(\lambda)| \le -\tan(\alpha)\mathrm{Re}(\lambda)\}$ for some $M < \infty$ and $0 < \alpha < \pi/2$. This assumption means that we are considering stiff ODEs and PDE diffusion–convection–reaction problems, where diffusion is not very weak. In fact, by using the model equation (3.1), we can investigate the behavior of a scheme on the entire spectrum of the operator in such a problem.

First, we know that the solution of (3.1) is $y(t) = y_0 e^{\lambda t}$, for $y(0) = y_0$. Consequently, the k-th derivative of the solution is $y^{(k)}(t) = \lambda^k y(t) = \lambda^k e^{\lambda t} y_0$. A simple calculus argument will prove the following proposition.

Proposition 3.1. *For any $T > 0$, the function*

$$g(\lambda) = \lambda^k e^{\lambda T}$$

is uniformly bounded for all $\lambda \in \mathcal{S}$. Namely, for $B = \tan(\alpha)$,

$$|g(\lambda)| \le \max\left\{ M^k e^{MT}, \left(\frac{k\sqrt{1+B^2}}{eT}\right)^k \right\}.$$

The uniform boundedness of $g(\lambda) = \lambda^k e^{\lambda k}$ can be understood as the smoothing property of the model equation $\dot{y} = \lambda y$. For any $\lambda \in \mathcal{S}$, the k-th derivative of the solution becomes uniformly bounded at any fixed positive time T.

A numerical scheme should also have a similar smoothing behavior. For the schemes in the Runge–Kutta family, it is well-known that a scheme applied to (3.1) can be written in the form of

$$y_{n+1} = R(\tau\lambda) y_n, \tag{3.2}$$

where $R(z)$ is called the stability function. The set of points z in the complex plane where $|R(z)| \le 1$ is called the stability region. Various notions of stability are defined on the basis of the properties of $R(z)$; see [5] among many other publications.

When $|R(z)| \le 1$ is satisfied, the scheme can still be "non-dissipative". The Crank–Nicolson scheme is well-known to be so. Therefore, it is necessary to find a way to identify those "dissipative" schemes. To this end, we investigate the general Runge–Kutta scheme solution. From (3.2),

$$y_n = [R(\tau\lambda)]^n y_0.$$

We want to consider the solution

$$\tilde{y}(t) = y_n e^{\lambda(t-t_n)},$$

because the smoothness of $\tilde{y}(t)$ in $(t_n, t_{n+1}]$ determines the actual local error $\tilde{y}(t_{n+1}) - y_N(t_{n+1})$, as shown in Fig. 2. Since

$$\tilde{y}^{(k)}(t) = y_n \lambda^k e^{\lambda(t-t_n)} = \lambda^k [R(\tau\lambda)]^n y_0 e^{\lambda(t-t_n)}, \tag{3.3}$$

taking into account that $n = T/\tau$, we define

$$S(\lambda, \tau, T) = \lambda^k [R(\tau\lambda)]^{\frac{T}{\tau}} \tag{3.4}$$

to be the **smoothing function**. Obviously, the behavior of the smoothing function tells us whether the scheme has any smoothing effect. If, for an arbitrarily given T, the smoothing function $S(\lambda, \tau, T)$ is bounded, then $\tilde{y}^{(k)}(t)$ is bounded in each step as shown in (3.3). Consequently, a k-th-order scheme can be used in the step.

In the case of solving the model equation for a given λ, one can actually compute $y_n \lambda^k$ as a **smoothing indicator**. (3.3) shows that the boundedness of $y_n \lambda^k$ implies the boundedness of $\tilde{y}^{(k)}(t)$ in (t_n, t_{n+1}). Moreover, the step size $\tau_n = t_{n+1} - t_n$ can be adaptively decided by using $y_n \lambda^k$ and a local error tolerance.

Definition. Let $\mathcal{R}$ be a subset of $\{(\tau, \lambda) : \tau > 0, \lambda \in \mathcal{S}\}$. If there is a positive continuous function $M(T)$ for $T > 0$, such that

$$|S(\tau, \lambda, T)| \le M(T)$$

for all $(\tau, \lambda) \in \mathcal{R}$, then we say that the scheme has a **smoothing effect** of order k in $\mathcal{R}$.

Remark. The stability condition $|R(\tau\lambda)| \le 1$ treats all values of the spectrum in the same way. We need to consider the uniform boundedness of $S(\tau, \lambda, T) = \lambda^k [R(\tau\lambda)]^{\frac{T}{\tau}}$, so that the lack-of-smoothing schemes can be identified. It seems impossible to define a unique smoothing region, unlike the concept of a stability region, because the region $\mathcal{R}$ usually depends on the **smoothing constant** $M(T)$. One can allow more or less smoothing by choosing $M(T)$ smaller or bigger, and consequently the region $\mathcal{R}$ becomes smaller or bigger correspondingly. □

4. Smoothing properties of Runge–Kutta schemes

First of all, we state and prove the following theorem.

Theorem 4.1. *For any given λ, if a scheme $y_{n+1} = R(\tau\lambda) y_n$ is consistent and stable, then it has a smoothing effect for sufficiently small step size τ.*

Proof. Since the scheme is consistent and stable, we have convergence, that is,

$$[R(\tau\lambda)]^{\frac{T}{\tau}} \to e^{\lambda T}$$

as $\tau \to 0$. Consequently,

$$S(\tau, \lambda, T) = \lambda^k [R(\tau\lambda)]^{\frac{T}{\tau}} \to \lambda^k e^{\lambda T},$$

which is known to be uniformly bounded as shown in Proposition 3.1 and its proof. □

Remark. Numerical smoothing is indeed a more restrictive condition compared to numerical stability, because it takes out the lack-of-smoothing or non-dissipative schemes among the stable schemes. However, the above theorem indicates the following. When we use a stable scheme, we achieve convergence only when we use a sufficiently small time step. With such a time step size, we actually have had numerical smoothing. □

It is well-known that the stability function for a Runge–Kutta scheme is a rational function; for $z = \tau\lambda$,

$$R(z) = \frac{p(z)}{q(z)}.$$

Therefore, we investigate the smoothing properties of Runge–Kutta schemes according to different categories of rational functions. The first case is when the degree of $p(z)$ is higher than the degree of $q(z)$.

Theorem 4.2. *If the degree of $p(z)$ is higher than the degree of $q(z)$, then:*

(a) *the stability region is bounded, where $|\tau\lambda| \le K$ for some $K < \infty$;*
(b) *in any compact subset of the stability region, which does not contain any point z where $|R(z)| = 1$, the scheme has a smoothing effect.*

Proof. Since the degree of $p(z)$ is higher than the degree of $q(z)$, when $|z| \to \infty$, $|R(z)|$ must be bigger than 1. Therefore the stability region must be bounded. If the stability region is in the disk $\{z: |z| \le K\}$, we have $|\tau\lambda| \le K$.

As for part (b), since $z = \tau\lambda$ is in a compact subset of the stability region where $|R(z)| \ne 1$, there must be a real number $\theta \in (0, 1)$ such that $|R(z)| \le \theta$. Now

$$|S(\tau, \lambda, T)| \le |\lambda|^k \theta^{\frac{T}{\tau}} \le |\lambda|^k \theta^{\frac{T|\lambda|}{K}}.$$

The right hand side is uniformly bounded for all $\lambda \in \mathcal{S}_\alpha$. □

Theorem 4.2 shows that an explicit or mostly explicit scheme almost always has a smoothing effect where it is stable. This includes the forward Euler scheme and classical Runge–Kutta schemes. If necessary, one just needs to shrink τ by a small percentage, to guarantee some smoothing. A smaller τ will make both θ and K smaller. Consequently the upper bound $M(T)$ of the smoothing function $S(\tau, \lambda, T)$ can be improved significantly.

In Theorem 4.2, as well as in the other theorems below, we focus on the case $\lambda \in \mathcal{S}_\alpha$. The case $\lambda \in \mathcal{S}_M$ is relatively easy to analyze for most schemes.

Next, we consider the implicit and mostly implicit schemes. The most popular implicit scheme is, of course, the backward Euler scheme.

Theorem 4.3. *The backward Euler scheme*

$$y_{n+1} = \frac{1}{1 - \tau\lambda} y_n$$

has a smoothing effect for any $\lambda \in \mathcal{S}_\alpha$ and any step size $\tau \le \frac{1}{M}$.

Proof. Without loss of generality, we can assume $T = km\tau$ for some integer m. Since $(1 + a)(1 + b) > 1 + a + b$ for any positive a and b, we have, for $\mathrm{Re}(\lambda) < 0$,

$$\frac{1}{|1 - \tau\lambda|^m} \le \frac{1}{|1 - \tau\mathrm{Re}(\lambda)|^m} < \frac{1}{1 - m\tau\mathrm{Re}(\lambda)} = \frac{1}{1 - \mathrm{Re}(\lambda)T/k}.$$

Hence

$$|S(\tau, \lambda, T)| = |\lambda|^k \left|\frac{1}{1 - \tau\lambda}\right|^{km} = \left|\frac{\lambda}{|1 - \tau\lambda|^m}\right|^k \le \left|\frac{k|\mathrm{Re}(\lambda)|\sqrt{1 + B^2}}{k - T\mathrm{Re}(\lambda)}\right|^k \le \left|\frac{k\sqrt{1 + B^2}}{T}\right|^k,$$

which is uniformly bounded for all $\lambda \in \mathcal{S}_\alpha$. □

Next, we consider other mostly implicit schemes. In this case, we give a simple result when $|\tau\lambda|$ is sufficiently large. Although this does imply that the schemes of this category are "unconditionally smoothing" according to our definition, the smoothing constant $M(T)$ that we would get from this analysis will not be useful in practice. Further study is certainly needed for each concrete scheme, where much better smoothing constants can be obtained by using detailed information about the scheme.

Theorem 4.4. *If the degree of $p(z)$ is lower than the degree of $q(z)$ in the scheme*

$$y_{n+1} = R(\tau\lambda)y_n,$$

then the scheme has a smoothing effect when $-\mathrm{Re}(z)$ is sufficiently large.

Proof. It just takes some elementary algebra and inequalities to show that, when the degree of $p(z)$ is lower than the degree of $q(z)$ and $-\mathrm{Re}(z)$ is sufficiently large, there is a constant $\beta > 0$, such the $|R(z)| \le \frac{1}{1-\beta\mathrm{Re}(z)}$. Once this is done, we just need to use the same argument as in the proof of Theorem 4.3. □

The last case is when the degree of $p(z)$ is equal to the degree of $q(z)$. There are three sub-cases here. First, if $\lim_{\mathrm{Re}(z)\to-\infty}|R(z)| > 1$, then the stability region must be bounded, and the result of Theorem 4.2 applies. Second, if $\lim_{\mathrm{Re}(z)\to-\infty}|R(z)| = \theta < 1$, then for sufficiently large $-\mathrm{Re}(z)$,

$$|S(\tau,\lambda,T)| \cong |\lambda|^k\theta^{\frac{T}{\tau}}.$$

For this smoothing function to be bounded by a constant $M(T) < k\ln|\lambda|$, we need to have, after a simple derivation,

$$\tau \le \frac{T\ln(1/\theta)}{k\ln|\lambda| - \ln M(T)}.$$

Here we obviously do not have unconditional smoothing, although the dependence of τ on λ is very mild.

The last sub-case is when $\lim_{\mathrm{Re}(z)\to-\infty}|R(z)| = 1$. In this article we only discuss the most popular scheme of this category, namely, the Crank–Nicolson scheme,

$$y_{n+1} = \frac{1+\tau\lambda/2}{1-\tau\lambda/2}y_n.$$

Here we have the following theorem.

Theorem 4.5. (a) *If $\lambda < 0$ and $\tau = c/\sqrt{|\lambda|}$ for some constant $c > 0$, the Crank–Nicolson scheme **does not have smoothing**.*
(b) *If $\tau \le \frac{c}{A^p}$ for $\lambda = -A + \mathrm{i}BA$ and some constants $c > 0$, $p > \frac{1}{2}$ and $B = \tan(\alpha)$, the Crank–Nicolson scheme has a smoothing effect, with a smoothing constant $M(T)$ independent of A.*

Proof. For the proof of (a),

$$\left|\frac{1+\tau\lambda/2}{1-\tau\lambda/2}\right|^{\frac{T}{\tau}} = \left|\frac{1-\frac{2}{c\sqrt{|\lambda|}}}{1+\frac{2}{c\sqrt{|\lambda|}}}\right|^{\frac{T\sqrt{|\lambda|}}{c}} \to \mathrm{e}^{-\frac{4T}{c^2}}$$

as $\mathrm{Re}(\lambda)\to-\infty$. Since this limit is bounded away from 0 for any finite T, the smoothing function

$$S(\tau,\lambda,T) = \lambda^k\left|\frac{1+\tau\lambda/2}{1-\tau\lambda/2}\right|^{\frac{T}{\tau}}$$

is not uniformly bounded since it is asymptotic to $\lambda^k\mathrm{e}^{-\frac{4T}{c^2}}$.

For the proof of (b), we first notice that $\frac{\tau A}{1+\tau^2A^2(1+B^2)/4}$ attains its maximum $\frac{1}{\sqrt{1+B^2}} < 1$ at $\tau A = \frac{2}{\sqrt{1+B^2}}$. For $\tau A \in \left[\frac{2}{\sqrt{1+B^2}}, cA^{1-p}\right]$, it is easy to see that $\frac{\tau A}{1+\tau^2A^2(1+B^2)/4}$ decreases, and $1 < \tau^2A^2(1+B^2)/4$, so

$$\begin{aligned}
|S(\tau,\lambda,T)| &= |\lambda|^k\left|\frac{1+\tau\lambda/2}{1-\tau\lambda/2}\right|^{\frac{T}{\tau}}\\
&= A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\tau A+\tau^2A^2(1+B^2)/4}{1+\tau A+\tau^2A^2(1+B^2)/4}\right|^{\frac{T}{2\tau}}\\
&= A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{\tau A}{1+\tau^2A^2(1+B^2)/4}}{1+\frac{\tau A}{1+\tau^2A^2(1+B^2)/4}}\right|^{\frac{T}{2\tau}}
\end{aligned}$$

$$
\begin{aligned}
&\le A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{cA^{1-p}}{1+c^2A^{2-2p}(1+B^2)/4}}{1+\frac{cA^{1-p}}{1+c^2A^{2-2p}(1+B^2)/4}}\right|^{\frac{T}{2\tau}}\\
&\le A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{cA^{1-p}}{1+c^2A^{2-2p}(1+B^2)/4}}{1+\frac{cA^{1-p}}{1+c^2A^{2-2p}(1+B^2)/4}}\right|^{\frac{TA^p}{2c}}\\
&\le A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{cA^{1-p}}{2c^2A^{2-2p}(1+B^2)/4}}{1+\frac{cA^{1-p}}{2c^2A^{2-2p}(1+B^2)/4}}\right|^{\frac{TA^p}{2c}}\\
&= A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{2}{cA^{1-p}(1+B^2)}}{1+\frac{2}{cA^{1-p}(1+B^2)}}\right|^{\frac{cA^{1-p}(1+B^2)}{2}\frac{TA^{2p-1}}{c^2(1+B^2)}}\\
&\le A^k(1+B^2)^{\frac{k}{2}}\mathrm{e}^{-2\frac{TA^{2p-1}}{c^2(1+B^2)}},
\end{aligned}
$$

which is uniformly bounded for all A.

For $\tau A \in \left(0, \frac{2}{\sqrt{1+B^2}}\right]$, we have $\tau^2A^2(1+B^2)/4 \le 1$. So

$$
\begin{aligned}
|S(\tau,\lambda,T)| &= A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{\tau A}{1+\tau^2A^2(1+B^2)/4}}{1+\frac{\tau A}{1+\tau^2A^2(1+B^2)/4}}\right|^{\frac{T}{2\tau}}\\
&= A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{\tau A}{2}}{1+\frac{\tau A}{2}}\right|^{\frac{T}{2\tau}}\\
&= A^k(1+B^2)^{\frac{k}{2}}\left|\frac{1-\frac{\tau A}{2}}{1+\frac{\tau A}{2}}\right|^{\frac{1}{\tau A}\frac{TA}{2}}\\
&\le A^k(1+B^2)^{\frac{k}{2}}\mathrm{e}^{-\frac{TA}{2}},
\end{aligned}
$$

which is also uniformly bounded for all A. □

Remark. For a parabolic problem, where $\lambda = -C/h^2$ for the mesh size h, one should not use a time step size τ proportional to h ($\tau = ch$) while computing a transient process, because part (a) of the last theorem confirms that $\tau = ch$ gives no smoothing. It needs to be $\tau = ch^{1+\epsilon}$, with an $\epsilon > 0$, as indicated by part (b). According to the estimate at the end of the proof, a bigger $\epsilon = 2p - 1$ will improve the smoothing effect. Of course, it will also increase the cost of computation. In practice, an adaptive algorithm can automatically choose optimal step sizes.

References

[1] T. Sun, Stability and error analysis on partially implicit schemes, Numerical Methods for Partial Differential Equations 21 (2005) 843–858.
[2] R.D. Richtmyer, K.W. Morton, Difference Methods for Initial-Value Problems, second ed., Interscience Publishers, New York, 1967.
[3] T. Sun, R. Ewing, Long-time error estimation and a stability indicator, Dynamics of Continuous, Discrete and Impulsive Systems. Series B. Applications and Algorithms 9 (2002) 115–129.
[4] T. Sun, D. Fillipova, Long-time error estimation on semi-linear parabolic equations, Journal of Computational and Applied Mathematics 185 (2006) 1–18.
[5] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff problems, Springer-Verlag, New York, 1987.
[6] P.D. Lax, R.D. Richtmyer, Survey of the stability of linear finite difference equations, Communications on Pure and Applied Mathematics 9 (1956) 267–293.