Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 193 (2006) 614–637

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Numerical method of estimating the blow-up time and rate of the solution of ordinary differential equations—An application to the blow-up problems of partial differential equations

Chiaki Hirota*, Kazufumi Ozawa

Faculty of Systems Science and Technology, Akita Prefectural University, Yurihonjo City, Akita 015–0055, Japan

Received 20 July 2004

Dedicated to Professor Tom Mitsui on the occasion of his 60th birthday

Abstract

A numerical method is proposed for estimating the blow-up time and the blow-up rate of the solution of ordinary differential equation (ODE), when the solution diverges at a finite time, that is, the blow-up time. The main idea is to transform the ODE into a tractable form by the *arc length transformation* technique [S. Moriguti, C. Okuno, R. Suekane, M. Iri, K. Takeuchi, Ikiteiru Suugaku—Suuri Kougaku no Hatten (in Japanese), Baifukan, Tokyo, 1979.], and to generate a linearly convergent sequence to the blow-up time. The sequence is then accelerated by the Aitken Δ^2 method. The present method is applied to the blow-up problems of partial differential equations (PDEs) by discretising the equations in space and integrating the resulting ODEs by an ODE solver, that is, the method of lines approach. Numerical experiments on the three PDEs, the semi-linear reaction–diffusion equation, the heat equation with a nonlinear boundary condition and the semi-linear reaction–diffusion system, show the validity of the present method.

Keywords: Blow-up time; Blow-up problems of PDEs; Parabolic equations; Method of lines; Arc length transformation; Aitken Δ^2 method; Linearly convergent sequence

* Corresponding author. Fax: +81 184 27 2187.

E-mail addresses: chiaki@akita-pu.ac.jp (C. Hirota), ozawa@akita-pu.ac.jp (K. Ozawa).

0377-0427/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.cam.2005.04.069

1. Introduction

It is often the case in practice that the solution of ordinary differential equation (ODE) or partial differential equation (PDE) diverges at a finite time. Such a phenomenon is said to be blow-up phenomenon. There exist many differential equations with blow-up solutions, such as the semi-linear reaction–diffusion equation with a nonlinear reaction term, the heat equation with a nonlinear boundary condition, and the nonlinear Schrödinger equation. The blow-up phenomena arise in various fields of science. For example, combustion in chemistry [8,25,28], curvature flow in geometry [6] and chemotaxis in biology [28]. Therefore, the mathematical analysis of the blow-up phenomena gives an interesting application to those fields.

There are many mathematical researches on the blow-up phenomena, including the condition for the solution to blow up, the blow-up set, i.e. the set of points at which the solution becomes unbounded, the blow-up time, and the asymptotic rate of divergence (blow-up rate). For the blow-up conditions, the conditions for the solutions of semi- or fully discretised equations of PDEs to blow up are derived (see e.g. [1,3,9,10,12,23]). For the blow-up sets, Fernández Bonder et al. [15] and Groisman and Rossi [18] investigate the relation between the blow-up sets of the continuous and the semi-discretised equations, and show the convergence of the blow-up set of the semi-discretised equation to that of the continuous model as the spatial mesh-size approaches 0. In this situation, the convergence of the blow-up times of the semi-discretised equations to those of the parabolic equations are also established by Abia et al. [1,2] and Ushijima [30]. Finally, for the blow-up rate, Abia et al. [1] and Acosta et al. [5] investigate the relation between the rate of original problem and that of the semi-discretised ones. On the other hand, Ishiwata and Yazaki [20] give characterisations of the blow-up solutions using the blow-up rates. An excellent survey on the blow-up solution of diffusion equations has been published [7]. In the survey, the authors emphasise the importance of numerical studies in this area, since analytical tools for the blow-up time and rate have not yet been given sufficiently.

In this paper, we will give an efficient numerical algorithm for estimating the blow-up time of the solution of ODEs, and then apply the algorithm to the blow-up problem of parabolic PDEs. The organization of this paper is as follows: in Section 2 three typical blow-up problems of PDEs are presented. In Section 3, we will give an efficient algorithm to estimate the blow-up time of ODE blow-up problems based on the main theorem (Theorem 9) which states: let $y(t)$ be the solution of the ODE which diverges at time $t = T(< +\infty)$ with the asymptotic order $y(t) \sim (T-t)^{-p}(p > 0)$, and s be the arc length of the solution, then the sequence $t(s_l)(l = 0, 1, \ldots)$, the value of t at $s = s_l$, is a linearly convergent sequence to T, when s_l is a geometric sequence diverging to $+\infty$. In the proposed algorithm, the original equation is transformed into a numerically tractable one, in which the independent variable is the arc length of the original equation and t is one of the dependent variables. We integrate the transformed equation and extract the linearly convergent sequence from the values of t. The sequence is accelerated by the Aitken Δ^2 method. In Section 4, the algorithm is applied to the three blow-up problems described in Section 2. Section 5 is the conclusion.

2. PDEs with blow-up solutions

We are mainly concerned with the three initial-boundary value problems of PDEs, which are the most famous blow-up problems.

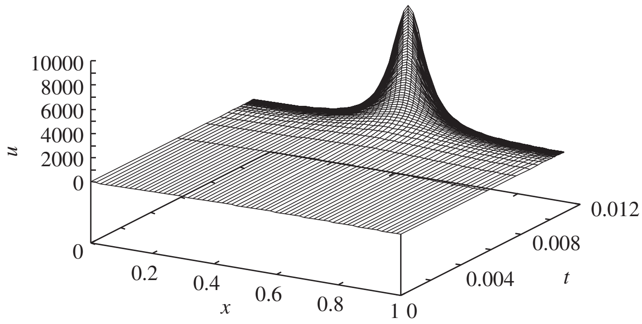

Fig. 1. Solution profile of (1) when $\Omega = (0, 1)$, $f(u) = u^2$ and $u^0(x) = 100 \sin \pi x$.

Problem 1 (*Semi-linear reaction–diffusion equation*). *The first problem is*

$$\begin{aligned} &\frac{\partial u}{\partial t} = \Delta u + f(u), \quad (x, t) \in \Omega \times (0, T), \\ &u(x, t) = 0, \quad x \in \partial\Omega, \ t \in [0, T), \\ &u(x, 0) = u^0(x), \quad x \in \Omega, \end{aligned} \tag{1}$$

where Ω is a bounded domain in $\mathbb{R}^d$, and the initial function u^0 is a smooth, non-trivial and non-negative function satisfying $u^0(x) = 0$ $(x \in \partial\Omega)$.

When the reaction term is $f(u) = u^r$ $(r > 1)$ or $f(u) = e^u$, Fujita [17] has shown the existence of the blow-up solution, i.e. for a finite time $0 < T < \infty$ the solution u satisfies

$$\lim_{t \uparrow T} \|u(\cdot, t)\|_\infty = \infty,$$

if the initial function u^0 is sufficiently large. For the case that $\Omega = (0, 1)$ and $f(u) = u^2$, the numerical solution corresponding to $u^0(x) = 100 \sin \pi x$ is shown in Fig. 1. From the figure we can observe the rapidly growing behaviour of the solution.

For the problem with $f(u) = u^2$, Nakagawa [23] has proposed a fully discretised scheme, and Chen [9,10] has extended the Nakagawa's scheme. We explain these schemes in Appendix. These schemes have been derived under the assumption that the solution blows up in L^q-norm ($q = 1$ or 2). However, Friedman and McLeod [16] pointed out the existence of the reaction term f which does not satisfy the assumption. Thus, the applicability of these fully discretised schemes is considerably restricted.

On the other hand, Abia et al. [1,2] have suggested to use the semi-discretised equation

$$\begin{aligned} &\frac{\mathrm{d}U_i}{\mathrm{d}t} = \delta^2 U_i + f(U_i), \quad i = 1, \ldots, n-1, \ t > 0, \\ &U_0(t) = U_n(t) = 0, \quad t > 0, \\ &U_i(0) = u^0(x_i), \quad i = 0, \ldots, n, \end{aligned} \tag{2}$$

where $U_i(t)$ is an approximation to $u(i\Delta x, t)$, and $\delta^2 U_i$ is the standard second order difference approximation to Δu at $x = i\Delta x$. They have also shown for one-dimensional case that the blow-up time of the

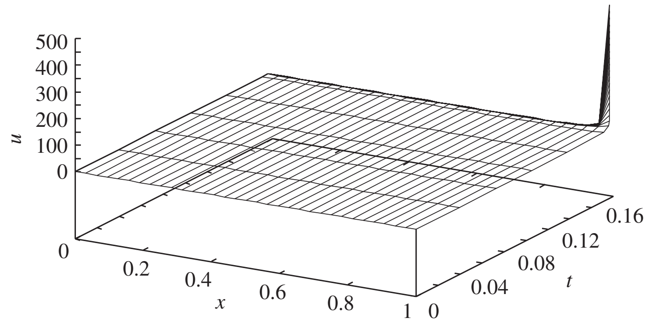

Fig. 2. Solution profile of (3) when $\Omega = (0, 1)$, $f(u) = u^2$ and $u^0(x) = 1$.

solution of (2) converges to that of (1) as $\Delta x \to 0$. This result is extended by Ushijima [30] to the multi-dimensional cases. Therefore, to analyse Eq. (1) with more general reaction terms, it is advantageous to use the semi-discretised equation instead of the fully discretised one.

Problem 2 (*Heat equation with a nonlinear boundary condition*). *The second problem to be considered is*

$$\begin{aligned}
&\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad (x,t) \in (0,1) \times (0,T),\\
&\left.\frac{\partial u}{\partial x}\right|_{x=1} = f(u(1,t)), \quad t \in [0,T),\\
&\left.\frac{\partial u}{\partial x}\right|_{x=0} = 0, \quad t \in [0,T),\\
&u(x,0) = u^0(x) \geqslant 0, \quad x \in [0,1],
\end{aligned} \tag{3}$$

where $f > 0$ is a smooth and increasing function, and u^0 is a non-zero function satisfying the boundary condition.

This equation is also shown to have a blow-up solution under certain conditions [31]. For example, for the case that $f(u) = u^2$, if the initial function is $u^0(x) = 1$, the solution blows up. The numerical solution for this case is shown in Fig. 2. Also for Eq. (3), the semi-discretised equation has frequently been used to analyse the blow-up phenomenon [5,4,15,24], and the blow-up time of the equation is shown to converge to that of the solution of (3), as the spatial mesh-size approaches 0 [12].

Problem 3 (*Semi-linear reaction–diffusion system [13]*). *The third problem is*

$$\left.\begin{aligned}
&\frac{\partial u}{\partial t} = \Delta u + v^{r_1},\\
&\frac{\partial v}{\partial t} = \Delta v + u^{r_2},
\end{aligned}\right. \quad (x,t) \in \mathbb{R}^d \times (0,T), \tag{4}$$

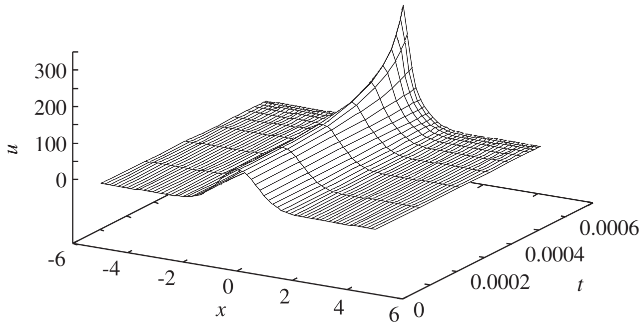

Fig. 3. An example of the solution $u(x,t)$ of Eq. (4) for one-dimensional case.

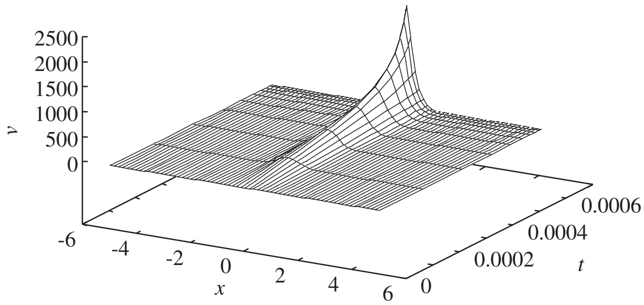

Fig. 4. An example of the solution $v(x,t)$ of Eq. (4) for one-dimensional case.

$$u(x,0)=u^0(x), \quad v(x,0)=v^0(x),$$
$$\lim_{\|x\|\to\infty} u(x,t)=\lim_{\|x\|\to\infty} v(x,t)=0.$$

Non-trivial solutions u and v of the problem blow up, when $r_1r_2>1$ and

$$\frac{r+1}{r_1r_2-1}\geqslant\frac{d}{2}, \tag{5}$$

where $r=\max\{r_1,r_2\}$ (see [13]). To analyse the blow-up phenomena of the equation, we also use the semi-discretised equation. For the one-dimensional case that

$$r_1=2, \quad r_2=3, \quad u^0(x)=100\,\exp(-x^2), \quad v^0(x)=0,$$

the solution profiles are shown in Fig. 3 and 4.

Since we have decided to use the semi-discretised equations instead of the fully discretised ones, we are now in a position to consider the problem of estimating the blow-up time of the solution of the system of ODEs.

3. Blow-up time and blow-up rate of the solution of ODE

Also for the solution of the ODE: $y'(t) = f(y)$, if $|y(t)|$ or $\|y(t)\|$ diverges at a finite time $t = T$, then y and T are called the blow-up solution and the blow-up time, respectively. As an example of the ODEs with blow-up solutions, consider the following initial value problem of scalar ODE:

Problem 4 (*A simple ODE with blow-up solution*).

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y^{\alpha}, \quad t > 0, \quad \alpha > 1,$$
$$y(0) = 1. \tag{6}$$

The exact solution is

$$y(t) = \frac{1}{\{1 - (\alpha - 1)t\}^{1/(\alpha-1)}}, \tag{7}$$

so that the solution has a pole of order $1/(\alpha - 1)$ *at* $t = 1/(\alpha - 1)$.

For a while, we consider the problem of estimating the blow-up time for this equation.

3.1. A simple method for estimating the blow-up time

At first we can think of a simple method: let us take a sufficiently large constant $M > 0$, and compute the numerical solutions $y_1, y_2, \ldots$, which correspond to the exact solutions at the points $t = t_1 < t_2 < \cdots$, respectively. Suppose that the numerical solution grows up gradually, and at last the condition $|y_m| > M$ or $\|y_m\| > M$ is satisfied for some m. Then stop the computation immediately, and take the value $t = t_m$ as an approximation to the blow-up time $T = 1/(\alpha - 1)$.

Let us apply this simple method to Eq. (6) with $\alpha = 2$. We set $M = 10^5$ and use the forward Euler method. The obtained approximations for the step-sizes $\Delta t = 0.1, 0.05$ and 0.02 are 1.6, 1.3 and 1.14, respectively. The numerical solutions, together with the exact solution, are shown in Fig. 5. As might be expected, the numerical solutions appear even in the region $t \geqslant 1$, where the exact solution is not defined. The error analysis of this simple method when applied to Eq. (6) is given by Sanz-Serna and Verwer [26]. We explain this analysis briefly.

Let t^* be the point $t_m = m\Delta t$ (m is not necessarily an integer) at which the linearly interpolated value of the numerical solution (the Euler polygon) satisfies $y_m = M$. Then Sanz-Serna and Verwer [26] gives

$$t^* - T = -\frac{1}{(\alpha - 1)M^{\alpha-1}} + \frac{\alpha\Delta t}{2}\log M + \mathrm{O}((\Delta t)^2). \tag{8}$$

Here, if we set

$$\Delta t = \frac{2}{\alpha(\alpha - 1)M^{\alpha-1}\log M}, \tag{9}$$

then we have

$$t^* - T = \mathrm{O}((\Delta t)^2).$$

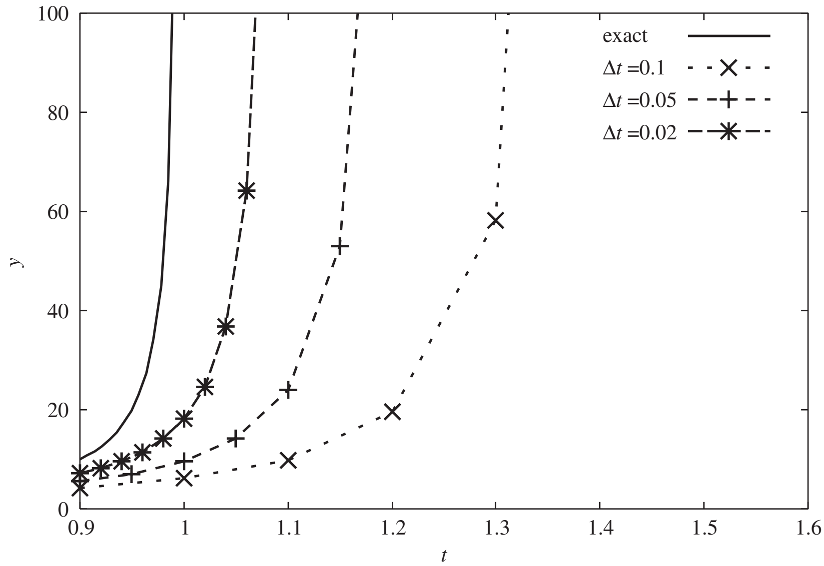

Fig. 5. Exact and numerical solutions of (6) with $\alpha = 2$, where the numerical method is the explicit Euler method with the step-sizes 0.1, 0.05, and 0.02.

Relation (9) implies that the larger the value of M, the smaller the step-size Δt, particularly when α is large, although it is necessary to take a large M to raise the accuracy of the estimated blow-up time.

Although the experiment and the error analysis above are for the forward Euler method, which is the most primitive numerical method, even for more sophisticated methods the same difficulty would be expected, since conventional numerical methods are designed without taking into account the singularity such as blow-up. On the other hand, several numerical methods were proposed to deal with the singularity (see [14, pp. 125–139], [21, pp. 209–216]). These special methods, however, were not for systems of equations but for scalar equations, so that these methods cannot be applied to the semi-discretised equations derived from PDEs. Thus, the best way to analyse the blow-up phenomena is to eliminate the singularity by a change of variables, and to keep the magnitudes of the values appearing in the computation as small as possible to avoid overflows.

3.2. Arc length transformation

As a change of variables which eliminates the singularity of the blow-up solution, two methods are currently known. The first one is proposed by Acosta et al. [4]. They have suggested to solve the reciprocal equation

$$\frac{\mathrm{d}t}{\mathrm{d}y} = \frac{1}{f(y)}, \qquad y \in (y^0, \infty),$$

instead of $\mathrm{d}y/\mathrm{d}t = f(y)$. In the case of a system of ODEs, we must choose the most rapidly growing component of y as the independent variable. In practice, however, we will seldom know such a component in advance. Therefore, the method has very limited applicability.

The second method is the one proposed by Moriguti [22], which can be used without knowing which component blows up, and therefore easily applied to systems of ODEs. Consider the following initial value problem of the system of ODEs:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y(t)), \quad 0 < t < T,$$
$$y(0) = y^0, \tag{10}$$

where $y = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$, $f(y) = (f_1(y), f_2(y), \ldots, f_n(y))^{\mathrm{T}}$ and $y^0 = (y_1^0, y_2^0, \ldots, y_n^0)^{\mathrm{T}}$. Hereafter, we regard the variables t and y_i as functions of the *arc length* s. Since $\mathrm{d}s^2 = \mathrm{d}t^2 + \mathrm{d}y_1^2 + \cdots + \mathrm{d}y_n^2$, the variables $t(s)$ and $y_i(s)$ satisfy the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}s}\begin{pmatrix} t \\ y_1 \\ \vdots \\ y_n \end{pmatrix} = \frac{1}{\sqrt{1 + \sum_{i=1}^n f_i^2}} \begin{pmatrix} 1 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}, \quad 0 < s < \infty,$$
$$t(0) = 0, \quad y(0) = y^0. \tag{11}$$

We call this transformation *arc length transformation*. Note that in the transformed equation (11), the solution y never blows up for finite s. Thus, the difficulty described in the previous section never arises. Moreover, when t approaches T (s approaches $+\infty$) each component of the right-hand side of (11) satisfies

$$\lim_{t \to T} \frac{f_k}{\sqrt{1 + \sum_{i=1}^n f_i^2}} = \begin{cases} \text{const } (\neq 0) & \text{if } k \in \mathscr{B}, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where

$$\mathscr{B} = \{k \mid y_k \text{ blows up with highest order}\}. \tag{13}$$

Therefore, the original equation is transformed into a numerically tractable one. To show this visually, consider Eq. (6) with $\alpha = 2$. The equation is transformed into

$$\frac{\mathrm{d}}{\mathrm{d}s}\begin{pmatrix} t \\ y \end{pmatrix} = \frac{1}{\sqrt{1 + y^4}} \begin{pmatrix} 1 \\ y^2 \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} g_0 \\ g_1 \end{pmatrix}.$$

The behaviours of g_0 and g_1, which denote the gradients of the solutions t and y, together with the behaviours of the solutions are shown in Figs. 6 and 7. We can find from the figures that g_0 and g_1 approach 0 and 1, respectively, as $s \to +\infty$.

Next, in order to see how well the arc length transformation works, we will practically estimate the blow-up time of the solution of the ODE (6) by applying the simple method in the previous sub-section. We can easily understand from Figs. 6 and 7 that considerably larger step-size will be permitted when s is large. Therefore, we use the adaptive code for the integration of the ODEs.

We use the `DOPRI5` [19], which is one of the famous ODE codes, throughout our experiments, since the extensive study by Hairer et al. [19] shows that the code is most efficient among the codes of the same order. Here we briefly explain the `DOPRI5`. This code has been written by Hairer and Wanner based on the pair of the 4th and 5th order explicit Runge–Kutta methods by Dormand and Prince [11]. In the code

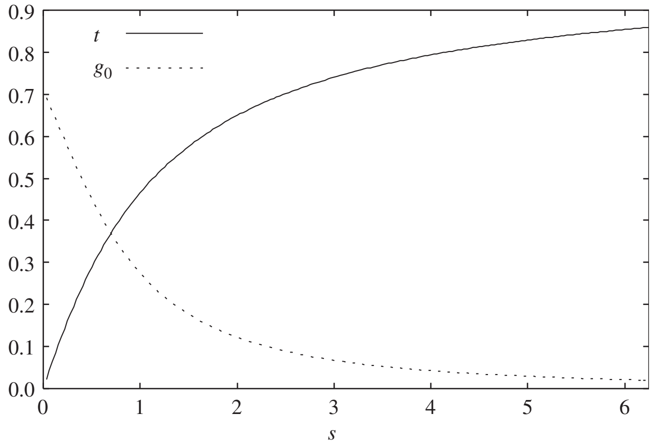

Fig. 6. $t(s)$ and $g_0(s)$ versus s for (6) with $\alpha = 2$.

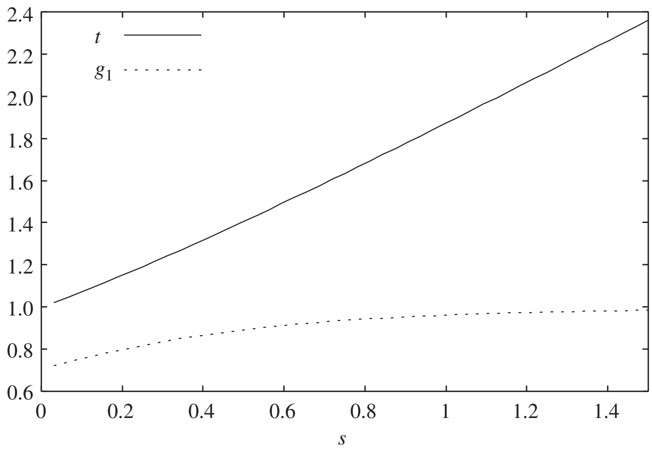

Fig. 7. $y(s)$ and $g_1(s)$ versus s for (6) with $\alpha = 2$.

the higher order method is used to carry the numerical solution, and the lower order method, together with the higher order one, is used to control the step-size so as to keep the local error within the user-specified tolerances. For the tolerances, the code has three parameters, `ATOL`, `RTOL` and `ITOL`. The parameters `ATOL` and `RTOL` specify the tolerances of the absolute and relative errors, respectively, and `ITOL` is used to choose the manner in which the errors are controlled; if we set `ITOL` = `0` then the error is controlled in normwise manner, and if we set `ITOL` = `1` then the error is controlled in componentwise manner. In our experiments we set `ATOL` = `RTOL` = `1.d-15` and `ITOL` = `0`.

The experimental results for Problem 4 with the two parameters in Table 1 are shown in Tables 2 and 3. In these experiments the double precision IEEE arithmetic is used. From the results, we can find that the values of s become very large even for the transformed equation, although the arc length transformation reduces the computational work tremendously.

Table 1
Examples of Problem 4

α	Equation	Initial value	Solution	Blow-up time T	Blow-up rate
2	$y' = y^2$	$y^0 = 1$	$y(t) = \frac{1}{1-t}$	1	1
$\frac{11}{10}$	$y' = y^{11/10}$	$y^0 = 1$	$y(t) = \frac{1}{(1-\frac{t}{10})^{10}}$	10	10

Table 2
Eq. (6) with $\alpha = 2$

$\log_{10} M$	Untransformed eq.		Transformed eq.		
	$t_m - T$	Steps	s	$t_m - T$	Steps
4	−9.994d − 05	4178	1.014d + 04	−9.857d − 05	1079
5	−9.995d − 06	5240	1.024d + 05	−9.767d − 06	1157
6	−9.997d − 07	6302	1.005d + 06	−9.950d − 07	1207
7	−9.999d − 08	7364	1.080d + 07	−9.261d − 08	1241
8	−9.979d − 09	8427	1.095d + 08	−9.135d − 09	1263
9	−9.980d − 10	9489	1.109d + 09	−9.019d − 10	1278
10	−9.982d − 11	10551	1.279d + 10	−7.814d − 11	1289
11	−9.980d − 12	11613	1.104d + 11	−9.041d − 12	1296

Steps: the number of the steps in the `DOPRI5` code.
t_m: The first t-value at which $|y_m| > M$.

Table 3
Eq. (6) with $\alpha = 11/10$

$\log_{10} M$	Untransformed eq.		Transformed eq.		
	$t_m - T$	Steps	s	$t_m - T$	Steps
10	−9.997d − 01	6906	1.005d + 10	−9.995d − 01	2841
20	−9.997d − 02	13856	1.003d + 20	−9.997d − 02	4470
30	−9.997d − 03	20806	1.005d + 30	−9.995d − 03	5503
40	−9.997d − 04	27756	1.039d + 40	−9.962d − 04	6167
50	−9.997d − 05	34706	1.048d + 50	−9.953d − 05	6597
60	−9.998d − 06	41656	1.104d + 60	−9.902d − 06	6880
70	−9.998d − 07	48606	1.125d + 70	−9.883d − 07	7069

Steps: the number of the steps in the `DOPRI5` code.
t_m: The first t-value at which $|y_m| > M$.

3.2.1. Linearly convergent sequence to the blow-up time and its acceleration

The algorithm to be proposed here generates a linearly convergent sequence to the blow-up time, and accelerates the sequence by the Aitken Δ^2 method for the case that the component $y_\nu(t)(\nu \in \mathscr{B})$ blows up in polynomial order. First of all, we assume the following:

Assumption 5.

- None of the components of $y(t)$ blows up anywhere in $[0, T)$, but at $t = T$ at least one component blows up (for brevity, we call the components which blow up the *blow-up components*).
- There may be many blow-up components, and they may tend to $+\infty$ or $-\infty$ with various orders. But the blow-up component(s) with highest order tends necessarily to $+\infty$, and is an increasing function on $t \in [0, T)$. That is, for all $v \in \mathscr{B}$

$$y_v(t) \to +\infty, \quad t \uparrow T \quad \text{and} \quad f_v(y(t)) > 0, \quad t \in [0, T).$$

From this assumption, we can easily show that the arc length s, which can be regarded as a function of t, also diverges at $t = T$, that is,

$$\lim_{t \uparrow T} s(t) = +\infty, \tag{14}$$

since for some $v \in \mathscr{B}$

$$s(t) = \int_0^t \sqrt{1 + \sum_{i=1}^n f_i^2}\, \mathrm{d}t > \int_0^t f_v \, \mathrm{d}t = y_v(t) - y_v(0) \to +\infty, \quad t \to T.$$

Moreover, the inverse function of $s(t)$, i.e. $t(s)$ exists and is single-valued for $s > 0$, since $\mathrm{d}s/\mathrm{d}t > 0$. Therefore, taking the inverse of (14) we have the following theorem:

Theorem 6. *Let us assume that the solution of* (10) *satisfies Assumption* 5, *then we have*

$$\lim_{s \to +\infty} t(s) = T.$$

This theorem guarantees the convergence of t to T for any blow-up solutions satisfying Assumption 5. On the other hand, for the divergence rate of s we have the following lemma:

Lemma 7. *Let $y(t)$ be the blow-up solution satisfying Assumption* 5, *and $z(s)$ be one of the blow-up components $y_v(t)$ for $v \in \mathscr{B}$, then*

$$\lim_{s \to +\infty} \frac{z(s)}{s} = \text{const.}$$

Proof. We have from (11)

$$z(s) = z(0) + \int_0^s \frac{g}{\sqrt{1 + \sum_{i=1}^n f_i^2}} \, \mathrm{d}s,$$

where we set $g = f_v$. By Assumption 5 this integrand approaches some positive constant, say G, as $s \to +\infty$. Therefore, if we put the integrand as

$$\frac{g}{\sqrt{1 + \sum_{i=1}^n f_i^2}} = G + \lambda(s),$$

then we have

$$\lim_{s\to+\infty} \lambda(s) = 0.$$

This means

$$\lim_{s\to+\infty} \frac{z(s)}{s} = \lim_{s\to+\infty} \left(\frac{z(0)}{s} + G + \frac{1}{s} \int_0^s \lambda(s)\,\mathrm{d}s \right) = G,$$

which completes the proof. □

We note here that the above results are independent of the rate of divergence. Hereafter, we will deal with the case that the blow-up component for $v \in \mathscr{B}$ has the following asymptotic property:

Assumption 8.

- For $v \in \mathscr{B}$ the blow-up component(s) y_v satisfies

$$y_v(t) \sim \frac{1}{(T-t)^p}, \quad t \uparrow T, \quad p > 0.$$

From this and former assumptions we have the following theorem:

Theorem 9. *Let the solution of* (10) *satisfy Assumption* 5 *and* 8, *and* $\{s_l\}$ *be the geometric sequence given by*

$$s_l = s_0 \cdot \gamma^l, \quad s_0 > 0, \quad \gamma > 1, \quad l = 0, 1, 2, \ldots .$$

Using the sequence, *if we define the sequence* $\{t_l\}$ *by*

$$t_l \stackrel{\text{def}}{=} t(s_l) = \int_0^{s_l} \frac{\mathrm{d}s}{\sqrt{1 + \sum_{i=1}^n f_i^2}}, \quad l = 0, 1, 2, \ldots, \tag{15}$$

then $\{t_l\}$ *converges to T linearly and the rate of convergence is* $\gamma^{-1/p}$.

Proof. Let $z(t)$ be one of $y_v(t)$'s for $v \in \mathscr{B}$ as before, and denote it by

$$z(t) = \frac{C}{(T-t)^p}(1 + \delta(T-t)), \tag{16}$$

where C is some positive constant and δ is a function satisfying

$$\lim_{\varepsilon\to+0} \delta(\varepsilon) = 0.$$

Using the new symbols ε_l and R_l given by

$$\varepsilon_l \stackrel{\text{def}}{=} T - t_l, \quad R_l \stackrel{\text{def}}{=} z(s_l)^{-1/p}, \quad l = 0, 1, 2, \ldots,$$

we have

$$\varepsilon_l = C^{1/p} R_l (1 + \delta(\varepsilon_l))^{1/p}.$$

Thus, we have from Lemma 7

$$\begin{aligned}\lim_{l\to\infty}\frac{\varepsilon_{l+1}}{\varepsilon_l} &= \lim_{l\to\infty}\left(\frac{R_{l+1}}{R_l}\right)\left(\frac{1+\delta(\varepsilon_{l+1})}{1+\delta(\varepsilon_l)}\right)^{1/p}\\ &= \lim_{l\to\infty}\left(\frac{R_{l+1}}{R_l}\right)\\ &= \lim_{l\to\infty}\left(\frac{z(s_l)}{z(s_{l+1})}\right)^{1/p}\\ &= \lim_{l\to\infty}\left(\frac{s_l}{s_{l+1}}\right)^{1/p}\\ &=\gamma^{-1/p}<1,\end{aligned}$$

which leads to the conclusion. □

This theorem shows that the convergence of $\{t_l\}$ is linear and the rate of convergence is unknown when $\{s_l\}$ is geometric. The following corollaries prove that the convergence of $\{t_l\}$ can be improved, if $\{s_l\}$ is faster than geometric.

Corollary 10. *Let* $\{\xi_l\}$ *be the sequence such that*

$$\xi_l>0,\quad \xi_l\to+\infty \quad \textit{and}\quad \frac{\xi_l}{\xi_{l+1}}\to 0,\quad l\to\infty.$$

Using the sequence, if we define

$$s_l=\xi_l\cdot\gamma^l,\quad \gamma>1,$$

then

$$\lim_{l\to\infty}\frac{\varepsilon_{l+1}}{\varepsilon_l}=0,$$

which shows the convergence of $\{t_l\}$ *to* T *is being superlinear.*

Corollary 11. *Let* $\{s_l\}$ *be the doubly exponential sequence given by*

$$s_l=K\gamma^{q^l},\quad K>0,\quad \gamma>1,\quad q>1,$$

then we have

$$\lim_{l\to\infty}\left|\frac{\varepsilon_{l+1}}{\varepsilon_l^q}\right|<\infty.$$

This means that the convergence of $\{t_l\}$ *to* T *is qth order.*

Above corollaries show that we can make the rate of convergence of $\{t_l\}$ arbitrarily fast by making the rate of divergence of $\{s_l\}$ arbitrarily fast. However, the faster the divergence rate of $\{s_l\}$, the higher the computational cost of $\{t_l\}$, since $\{t_l\}$ is obtained by integrating (11) from $s=0$ to s_l. Moreover, to avoid the danger of overflows in the floating-point arithmetic, it is advantageous not to use large numbers. Therefore, we decide to work with the linearly convergent sequence and to accelerate it. For the acceleration of linearly convergent sequence with unknown convergence rate, as in the present case, the Aitken Δ^2 method is a very useful tool [27,29]. We propose the following algorithm:

Algorithm.

(1) Let $s_0 > 0$ and $\gamma > 1$, and define the geometric sequence $\{s_l\}$ by

$$s_l = s_0 \cdot \gamma^l, \quad l = 0, 1, 2, \ldots .$$

(2) Integrate (11) from $s = 0$ to s_l and put $t_l = t(s_l)$.
(3) Let $t_l^{(0)} = t_l (l = 0, 1, 2, \ldots)$, and apply the Aitken Δ^2 method to the sequence recursively:

$$t_{l+2}^{(k+1)} = t_{l+2}^{(k)} - \frac{\left(t_{l+2}^{(k)} - t_{l+1}^{(k)}\right)^2}{t_{l+2}^{(k)} - 2t_{l+1}^{(k)} + t_l^{(k)}}, \quad l \geqslant 2k, \quad k = 0, 1, 2, \ldots$$

3.2.2. Estimation of the blow-up rate

Using our algorithm, we can also estimate the rate p, which we will call the *blow-up rate*. We will show the procedure.

Suppose that we have the sequences $\{t_l^{(k)}\}$ for $k = 0, \ldots, K$, $l = 2k, \ldots, L$ where $L \geqslant 2K$, and that the last value $t_L^{(K)}$ is satisfactory as an approximation of T. Then, we can expect for large l that

$$\lambda_l \stackrel{\text{def}}{=} \left| \frac{t_l^{(0)} - t_L^{(K)}}{t_{l-1}^{(0)} - t_L^{(K)}} \right| \simeq \gamma^{-1/p},$$

since the linear convergence of $\{t_l\}$ is established in Theorem 9. Thus, we expect

$$p_l \stackrel{\text{def}}{=} -1/\log_\gamma \lambda_l = -1/\log_\gamma \left| \frac{t_l^{(0)} - t_L^{(K)}}{t_{l-1}^{(0)} - t_L^{(K)}} \right| \tag{17}$$

to be a good approximant of the blow-up rate p.

3.3. Numerical experiments

It should be noted that the exact $\{t_l\}$ can never been obtained by our algorithm because of the roundoff and discretisation errors introduced in the computations. In order to obtain the nearly theoretical results, we use the double precision arithmetic and set the tolerance parameters very small.

We apply our algorithm to Problem 4, and estimate the blow-up time and rate. Now, let us define the sequence s_l by $s_l = 16 \cdot 2^l (l = 0, \ldots, 10)$, and the parameters in the `DOPRI5` be `ITOL` $=$ `0` and `ATOL` $=$ `RTOL` $=$ `1.d-15`, as before. The numerical results are shown in Tables 4 and 5. In the tables, $e_l^{(k)}$ denotes the error of $t_l^{(k)}$, that is,

$$e_l^{(k)} \stackrel{\text{def}}{=} t_l^{(k)} - T.$$

Table 4
Aitken Δ^2 process for Eq. (6) with $\alpha = 2$

l	s_l	$e_l^{(0)}$	$e_l^{(1)}$	$e_l^{(2)}$	$e_l^{(3)}$	p_l	Steps
0	16	−5.936d − 02					547
1	32	−3.044d − 02				1.038d + 00	640
2	64	−1.542d − 02	8.281d − 04			1.019d + 00	723
3	128	−7.761d − 03	2.069d − 04			1.010d + 00	796
4	256	−3.893d − 03	5.171d − 05	4.665d − 08		1.005d + 00	860
5	512	−1.950d − 03	1.293d − 05	2.613d − 09		1.002d + 00	916
6	1024	−9.758d − 04	3.232d − 06	1.528d − 10	7.284d − 12	1.001d + 00	965
7	2048	−4.881d − 04	8.080d − 07	9.212d − 12	3.093d − 13	1.001d + 00	1008
8	4096	−2.441d − 04	2.020d − 07	5.700d − 13	1.665d − 14	1.000d + 00	1046
9	8192	−1.221d − 04	5.050d − 08	4.663d − 14	1.288d − 14	1.000d + 00	1079
10	16384	−6.103d − 05	1.262d − 08	9.992d − 15	7.327d − 15	1.000d + 00	1108

Table 5
Aitken Δ^2 process for Eq. (6) with $\alpha = 11/10$

l	s_l	$e_l^{(0)}$	$e_l^{(1)}$	$e_l^{(2)}$	$e_l^{(3)}$	p_l	Steps
0	16	−7.550d + 00					485
1	32	−7.058d + 00				1.029d + 01	594
2	64	−6.591d + 00	1.877d + 00			1.014d + 01	701
3	128	−6.153d + 00	7.484d − 01			1.007d + 01	805
4	256	−5.742d + 00	3.258d − 01	7.292d − 02		1.003d + 01	906
5	512	−5.358d + 00	1.472d − 01	1.639d − 02		1.002d + 01	1004
6	1024	−5.000d + 00	6.763d − 02	3.705d − 03	3.644d − 05	1.001d + 01	1100
7	2048	−4.665d + 00	3.132d − 02	8.396d − 04	3.064d − 06	1.000d + 01	1194
8	4096	−4.353d + 00	1.456d − 02	1.907d − 04	7.491d − 07	1.000d + 01	1286
9	8192	−4.061d + 00	6.780d − 03	4.342d − 05	1.888d − 07	1.000d + 01	1376
10	16384	−3.789d + 00	3.160d − 03	9.913d − 06	4.281d − 08	1.000d + 01	1464

$e_l^{(k)} = t_l^{(k)} - T$, where T is the blow-up time, and $t_l^{(k)}$ is the sequence generated by the Aitken Δ^2 method.
p_l is the estimate of the blow-up rate (see Eq. (17)).
"Steps" denotes the number of the steps in the `DOPRI5` code.

The comparison between the results of Tables 2 and 3 shows that the present algorithm is effective, in particular, for the case $\alpha = \frac{11}{10}$. In the former experiment, the equation is being integrated until $s \simeq y > 10^{70}$ to obtain the result with the error of -9.883×10^{-7}, and the number of the steps is 7069. On the other hand, the application of the present algorithm reduces the steps to 1464, the error to 4.281×10^{-8}, and s-value to 16 384. Moreover, the estimates of the blow-up rate are found to be very accurate for both cases.

4. Application to the blow-up problems of PDEs

Next we consider the three blow-up problems of PDEs, Problems 1–3.

4.1. Semi-linear reaction–diffusion equation

Consider Problem 1 when $\Omega = (0, 1)$, $f(u) = u^2$ and $u^0(x) = 100 \sin \pi x$:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u^2, \quad (x, t) \in (0, 1) \times (0, T),$$
$$u(x, t) = 0, \quad x = 0, 1,$$
$$u(x, 0) = 100 \sin \pi x, \quad x \in [0, 1]. \tag{18}$$

Here, we divide the interval [0, 1] into n equi-length sub-intervals, i.e. we set $\Delta x = 1/n$, and denote by $U_i(t)$ the numerical approximation of $u(x, t)$ at $x = i\Delta x$. Using the standard central difference approximation, we have the system of ODEs:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{n-2} \\ U_{n-1} \end{pmatrix} = \frac{1}{(\Delta x)^2}\begin{pmatrix} -2 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -2 \end{pmatrix}\begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{n-2} \\ U_{n-1} \end{pmatrix} + \begin{pmatrix} U_1^2 \\ U_2^2 \\ \vdots \\ U_{n-2}^2 \\ U_{n-1}^2 \end{pmatrix}, \tag{19}$$

$$U_i(0) = u^0(i\Delta x), \quad i = 0, \ldots, n.$$

This solution has the blow-up rate 1, since for Problem 1 with $f(u) = u^r\,(r > 1)$, Groisman and Rossi [18] has shown

$$\max_i U_i(t) \sim (T - t)^{-1/(r-1)}, \quad t \uparrow T. \tag{20}$$

This means that Assumption 8 is satisfied with $p = 1$.

We apply our algorithm to (19) to estimate the blow-up time and rate. Here we set $n = 64$ and $s_l = 2^{16} \cdot 2^l$ $(l = 0, \ldots, 10)$, and accelerate the sequence four times by the Aitken Δ^2 method. The parameters in the `DOPRI5` are `ITOL = 0` and `ATOL = RTOL = 1.d-15`. The result is shown in Table 6. The table shows that the convergences of $t_l^{(k)}$ to T, which is unknown in this problem, appears to be valid, and that the estimated blow-up rate converges steadily to that given by (20).

To be more confident in our algorithm, next we perform the same experiment many times by halving the mesh size Δx repeatedly. Let T_n be the last value of the Aitken table, that is, the value in the last column and the last row in the table, when the mesh size is $\Delta x = 1/n$. If T_n is a sufficiently good approximation to the blow-up time of (19), then $T_{n/2} - T_n$ approximates the error in $T_{n/2}$. Therefore, if the value $|T_{n/2} - T_n|$ decreases with n at a proper rate, then we can ascertain the convergence of T_n to T. The values of $|T_{n/2} - T_n|$ for varying n are shown in Table 7. From the table, we can assure the convergence of T_n to the blow-up time of the solution of (18), since the rate of convergence is 2, which is just the accuracy of the difference approximation in space.

As is stated in Appendix, the blow-up time of the solution of (18) can be estimated also by Nakagawa's (26). We compare our scheme with Nakagawa's one with the same mesh size $\Delta x = \frac{1}{64}$. The parameter appearing in Nakagawa's scheme is set $\lambda = \frac{1}{2}$. The converging processes of $t_m^{(k)}$ together with that of the estimate by Nakagawa's scheme are shown in Figs. 8 and 9 (Fig. 9 is an enlargement of Fig. 8). These figures show that our algorithm is superior over Nakagawa's scheme.

Table 6
Blow-up time of the solution of Eq. (19) with $n = 64$

l	$t_l^{(0)}$	$t_l^{(1)}$	$t_l^{(2)}$	$t_l^{(3)}$	$t_l^{(4)}$	p_l	Steps
0	1.09516d − 02						1569
1	1.09699d − 02					7.795d − 01	1762
2	1.09774d − 02	1.09827d − 02				7.801d − 01	1956
3	1.09804d − 02	1.09824d − 02				8.219d − 01	2127
4	1.09816d − 02	1.09825d − 02	1.09825d − 02			8.899d − 01	2265
5	1.09822d − 02	1.09826d − 02	1.09822d − 02			9.415d − 01	2379
6	1.09824d − 02	1.09827d − 02	1.09827d − 02	1.09823d − 02		9.703d − 01	2476
7	1.09826d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02		9.851d − 01	2561
8	1.09826d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	9.925d − 01	2636
9	1.09826d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	9.963d − 01	2703
10	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	9.981d − 01	2764
11	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	9.991d − 01	2821
12	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	1.09827d − 02	9.995d − 01	2875

$t_l^{(k)}$ is the sequence generated by the Aitken Δ^2 method.
p_l is the estimate of the blow-up rate (see Eq. (17)).
"Steps" denotes the number of the steps in the `DOPRI5` code.

Table 7
Convergence behaviour of T_n to the blow-up time T of the solution of Eq. (18)

n	T_n	$\log_2 \lvert T_{n/2} - T_n \rvert$
16	1.095606426d − 02	
32	1.097700705d − 02	−15.54
64	1.098267421d − 02	−17.43
128	1.098417002d − 02	−19.35
256	1.098455990d − 02	−21.29
512	1.098465675d − 02	−23.30

T_n: Estimated blow-up time when the mesh size is $\Delta x = 1/n$.

4.2. Heat equation with nonlinear boundary condition

Next we consider Problem 2 when $f(u) = u^2$ and $u^0(x) = 1$:

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad (x, t) \in (0, 1) \times (0, T),\\
&\left.\frac{\partial u}{\partial x}\right|_{x=1} = u^2, \quad t \in [0, T),\\
&\left.\frac{\partial u}{\partial x}\right|_{x=0} = 0, \quad t \in [0, T),\\
&u(x, 0) = 1, \quad x \in [0, 1].
\end{aligned}
\tag{21}
$$

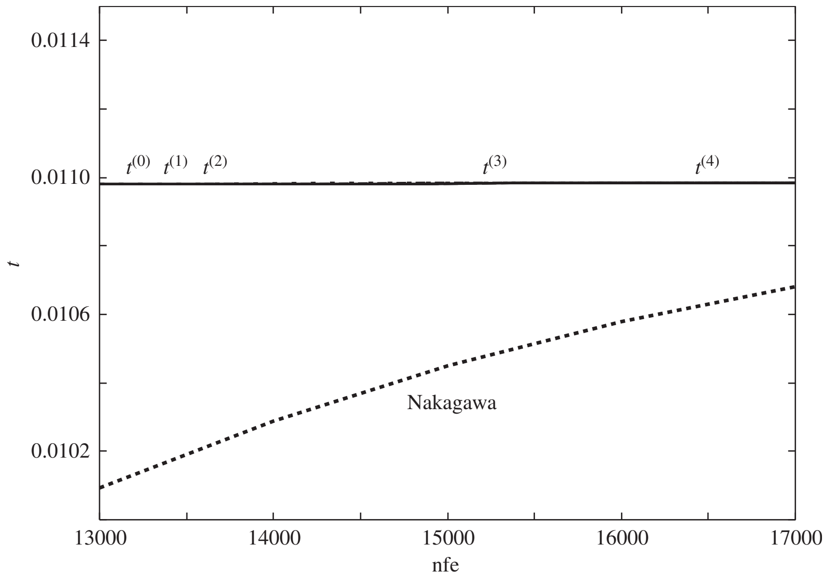

Fig. 8. Convergence to the blow-up time, where `nfe` denotes the number of function evaluations.

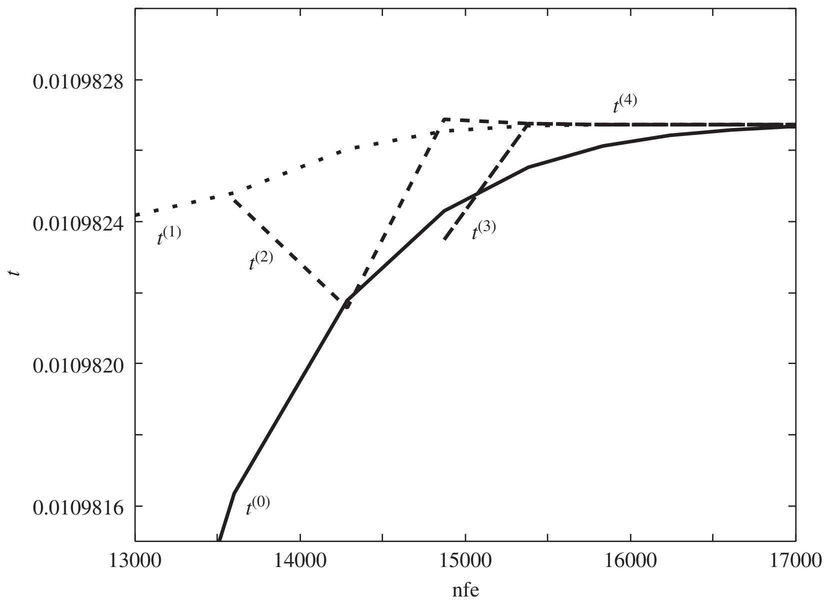

Fig. 9. Convergence to the blow-up time, where `nfe` denotes the number of function evaluations.

We use the central difference approximation for u_{xx}, as before. For the u_x's in the boundary conditions, we use the approximations $u_x(0,t) \simeq (U_1 - U_{-1})/(2\Delta x)$ and $u_x(1,t) \simeq (U_{n+1} - U_{n-1})/(2\Delta x)$, where U_{-1} and U_{n+1} are the approximations to the fictitious values $u(-\Delta x, t)$ and $u(1+\Delta x, t)$, respectively.

Table 8
Blow-up time of the solution of Eq. (22) with $n = 64$

l	$t_l^{(0)}$	$t_l^{(1)}$	$t_l^{(2)}$	$t_l^{(3)}$	$t_l^{(4)}$	p_l	Steps
0	1.76845d − 01						12799
1	1.76898d − 01					7.791d − 01	12908
2	1.76918d − 01	1.76931d − 01				8.752d − 01	13000
3	1.76927d − 01	1.76934d − 01				9.329d − 01	13086
4	1.76931d − 01	1.76935d − 01	1.76935d − 01			9.649d − 01	13168
5	1.76933d − 01	1.76935d − 01	1.76935d − 01			9.820d − 01	13247
6	1.76934d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01		9.908d − 01	13324
7	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01		9.954d − 01	13399
8	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	9.977d − 01	13472
9	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	9.988d − 01	13544
10	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	9.994d − 01	13615
11	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	9.997d − 01	13685
12	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	1.76935d − 01	9.999d − 01	13754

$t_l^{(k)}$ is the sequence generated by the Aitken Δ^2 method.
p_l is the estimate of the blow-up rate (see Eq. (17)).
"Steps" denotes the number of the steps in the `DOPRI5` code.

By equating the right-hand sides of these approximations to the corresponding values, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} U_0 \\ U_1 \\ \vdots \\ U_{n-1} \\ U_n \end{pmatrix} = \frac{1}{(\Delta x)^2}\begin{pmatrix} -2 & 2 & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 2 & -2 \end{pmatrix}\begin{pmatrix} U_0 \\ U_1 \\ \vdots \\ U_{n-1} \\ U_n \end{pmatrix} + \frac{2}{\Delta x}\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ U_n^2 \end{pmatrix}, \tag{22}$$

$$U_i(0) = u^0(i\Delta x), \quad i = 0, \ldots, n.$$

Also for this problem, Acosta et al. [5] have shown

$$\max_i U_i(t) \sim (T - t)^{-1/(r-1)}, \tag{23}$$

when $f(u) = u^r (r > 1)$. Since $r = 2$ in the present case, the blow-up rate of $U(t)$ at $t = T$ is 1, which means that Assumption 8 is satisfied with $p = 1$.

Here, we set $n = 64$ and $s_l = 2^7 \cdot 2^l$ $(l = 0, \ldots, 10)$. The parameters of `DOPRI5` are set `ITOL` = `0` and `ATOL` = `RTOL` = `1.d-15`, and the number of accelerations is 4, as before. The numerical result is shown in Table 8. From the table, we can conclude that the convergence of t_l to T, which is unknown also in this case, appears to be valid, and that the estimated blow-up rate coincides with that given by (23).

To be more confident in our algorithm, we perform again the same experiment as in the previous problem, and show the result in Table 9. In this table, we can observe approximately the same rate of convergence as that of the difference approximation used, and therefore we can confirm the validity of our algorithm also for this problem.

By the way, the solution of the problem is known to blow up at the boundary point (see e.g. [5]), so that the transformation technique by Acosta et al. [4], which has already been introduced in Section 3.2,

Table 9
Convergence behaviour of T_n to the blow-up time T of the solution of Eq. (21)

n	T_n	$\log_2\lvert T_{n/2} - T_n\rvert$
16	1.837915627d − 01	
32	1.785669900d − 01	−7.58
64	1.769353395d − 01	−9.26
128	1.764455048d − 01	−11.00
256	1.763025262d − 01	−12.77
512	1.762616488d − 01	−14.58
1024	1.762501461d − 01	−16.41
2048	1.762469497d − 01	−18.33
4096	1.762460703d − 01	−20.12

T_n: estimated blow-up time when the mesh size is $\Delta x = 1/n$.

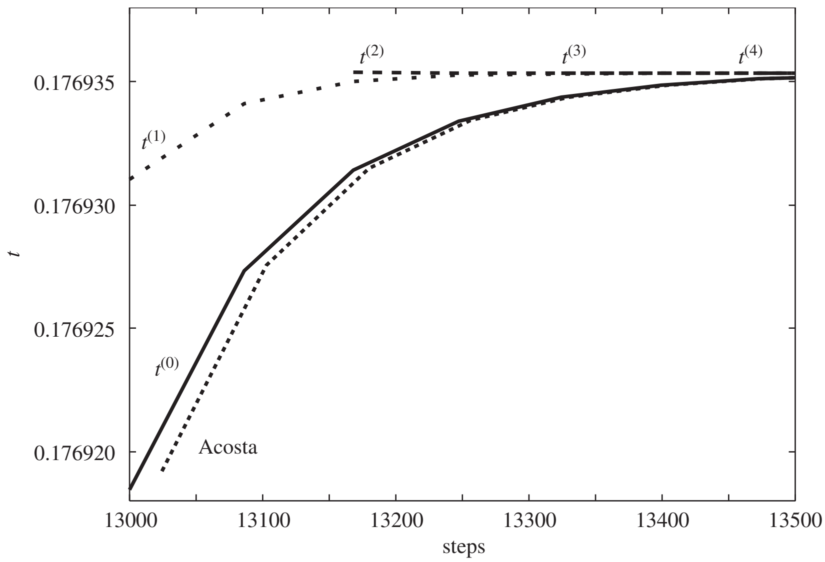

Fig. 10. Convergence behaviours of Acosta's and the present algorithms.

can be useful. The result of solving the transformed equation by the `DOPRI5` is illustrated and compared with that of our algorithm in Fig. 10. This figure shows the superiority of our method over the method of Acosta et al. [4] even without the accelerations.

4.3. Semi-linear reaction–diffusion system

As an example of Problem 3, consider the following problem:

Table 10
Estimation of the blow-up time of the solution of Eq. (25), where $n = 512$

l	$t_l^{(0)}$	$t_l^{(1)}$	$t_l^{(2)}$	$t_l^{(3)}$	$t_l^{(4)}$	p_l	Steps
0	6.55219d − 04						1680
1	6.55231d − 04					7.724d − 01	1787
2	6.55236d − 04	6.55239d − 04				7.861d − 01	1892
3	6.55238d − 04	6.55239d − 04				7.930d − 01	1996
4	6.55238d − 04	6.55239d − 04	6.55239d − 04			7.965d − 01	2099
5	6.55239d − 04	6.55239d − 04	6.55239d − 04			7.982d − 01	2202
6	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04		7.991d − 01	2305
7	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04		7.995d − 01	2408
8	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04	7.998d − 01	2511
9	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04	7.999d − 01	2614
10	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04	6.55239d − 04	7.999d − 01	2717

$t_l^{(k)}$ is the sequence generated by the Aitken Δ^2 method.
p_l is the estimate of the blow-up rate (see Eq. (17)).
"Steps" denotes the number of the steps in the `DOPRI5` code.

$$\left.\begin{aligned}\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + v^{r_1},\\ \frac{\partial v}{\partial t} &= \frac{\partial^2 v}{\partial x^2} + u^{r_2},\end{aligned}\right. \qquad (x,t) \in \mathbb{R} \times (0,T), \tag{24}$$

$$u(x,0) = 100\ \exp(-x^2), \quad v(x,0) = 0, \quad x \in \mathbb{R},$$

$$u(\pm\infty, t) = v(\pm\infty, t) = 0, \quad t \in [0,T).$$

For the numerical computation we consider the equation on the *finite interval* $x \in [-5, 5]$ instead of the whole space $\mathbb{R}$, and use the initial-boundary conditions

$$u(x,0) = 100\ \exp(-x^2), \quad v(x,0) = 0, \quad x \in (-5,5),$$

$$u(\pm 5, t) = v(\pm 5, t) = 0, \quad t \in [0,T),$$

instead of the exact conditions. Let n be the number of divisions in the interval, and u_i and v_i be approximations to the solutions at the ith point $x_i = -5 + i\Delta x$, where $\Delta x = 10/n$. The semi-discretised equation to be used is

$$\left.\begin{aligned}\frac{\mathrm{d}u_i}{\mathrm{d}t} &= \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} + v_i^{r_1},\\ \frac{\mathrm{d}v_i}{\mathrm{d}t} &= \frac{v_{i+1} - 2v_i + v_{i-1}}{(\Delta x)^2} + u_i^{r_2},\end{aligned}\right. \qquad i = 0, 1, \ldots, n, \tag{25}$$
$$u_i(0) = 100\ \exp(-x_i^2), \quad v_i(0) = 0.$$

Here we analyse the case $r_1 = 2$ and $r_2 = 3$, for which condition (5) holds. We apply our algorithm by setting $s_l = 2^{20} \cdot 2^l\,(l = 0, 1, \ldots, 10)$.

The result is given in Table 10. In the table the observed blow-up rate is approximately 0.8, which is just the value given by $(\max\{r_1, r_2\} + 1)/(r_1 r_2 - 1)$, the left-hand side of the inequality (5).

Table 11
Convergence behaviour of T_n to the blow-up time T of the solution of Eq. (24)

n	T_n	$\log_2\lvert T_{n/2} - T_n\rvert$
64	6.551052754d − 04	
128	6.551944188d − 04	−23.42
256	6.552278025d − 04	−24.84
512	6.552390173d − 04	−26.41
1024	6.552425366d − 04	−28.08
2048	6.552435922d − 04	−29.82
4096	6.552438991d − 04	−31.60

T_n: estimated blow-up time when the mesh size is $\Delta x = 10/n$.

As before, we denote by T_n the estimated blow-up time when the number of divisions in space is n. The values of T_n are shown in Table 11 for varying n. From the table, the convergence of T_n to T seems to be valid, although for this problem, the convergence has not yet been established theoretically. By the way, the cpu-time required to obtain the result on the PC with Pentium IV (2.4 GHz) is 40.6 s, when $n = 4096$.

Finally, we give some comments on our experiments. We have used very large values of s for the two reasons. Firstly, the arc length s becomes inevitably large for multi-dimensional cases. Secondly, in order to obtain nearly theoretical results, we have integrated the equation over fairly long ranges and chosen small values as the tolerance parameters. As a result, the results obtained for the blow-up problems of the PDEs are satisfactory, although the computation times become larger. Needless to say, if high accuracies are not required then we are able to relax the restrictions on the integration interval and the tolerance parameters.

5. Conclusion

We have proposed a method of estimating the blow-up time and the blow-up rate of the solution of the ODEs, and then applied the method to the blow-up problem of PDEs. The method is always applicable if the solution has a asymptotic property given by Assumption 8. To extend the present method to the solutions with another type of singularity such as that discovered by Angenent and Velázquez [6] will be a future work.

Acknowledgements

We would like to express our thanks to two professors, M. Okada (Tokyo Metropolitan University), and T. Ishiwata (Gifu University), for their useful suggestions, and to two anonymous referees, whose comments greatly improved this paper.

Appendix A. Nakagawa's scheme

Let us consider Problem 1 for the case that $\Omega = (0, 1)$ and $f(u) = u^2$. Then we denote by U_i^m the numerical approximation to the solution at the point (x_i, t_m), where x_i and t_m are the ith and mth grid points on the x- and t-axis, respectively. Nakagawa's scheme to compute U_i^m is given by

$$\frac{U_i^{m+1}-U_i^m}{\Delta t_m}=\frac{U_{i+1}^m-2U_i^m+U_{i-1}^m}{(\Delta x)^2}+(U_i^m)^2,\quad i=1,\ldots,n-1,$$
$$U_0^m=U_n^m=0,\quad m=0,1,2,\ldots,$$
$$\Delta x=\frac{1}{n}. \tag{26}$$

In this scheme, although the spatial mesh size Δx is fixed, the time step-size Δt_m is adjusted by the formula

$$\Delta t_m=t_m-t_{m-1}=\tau\times\min\left\{1,\frac{1}{\|U^m\|}\right\},\quad t_0=0,\quad \tau=\lambda(\Delta x)^2,$$

where λ is a predetermined constant satisfying $0<\lambda\leqslant\frac{1}{2}$, and $\|U^m\|$ is defined by

$$\|U^m\|=\left(\sum_{i=0}^{n}\Delta x\cdot(U_i^m)^2\right)^{1/2}.$$

Since $t_m\to T\,(m\to\infty)$ has been established in [23], we could estimate the blow-up time by pursuing the values of t_m, no matter what the efficiency is. Chen has extended Nakagawa's scheme to any case that $f(u)=u^r\,(r>1)$ [9], and further to the case that Ω is a closed ball in $\mathbb{R}^d$ [10].

References

[1] L.M. Abia, J.C. López-Marcos, J. Martínez, Blow-up for semidiscretization of reaction–diffusion equations, Appl. Numer. Math. 20 (1996) 145–156.
[2] L.M. Abia, J.C. López-Marcos, J. Martínez, On the blow-up time convergence of semidiscretizations of reaction–diffusion equations, Appl. Numer. Math. 26 (1998) 399–414.
[3] L.M. Abia, J.C. López-Marcos, J. Martínez, The Euler method in the numerical integration of reaction–diffusion problems with blow-up, Appl. Numer. Math. 38 (2001) 287–313.
[4] G. Acosta, G. Durán, J.D. Rossi, An adaptive time step procedure for a parabolic problem with blow-up, Computing 68 (2002) 343–373.
[5] G. Acosta, J. Fernández Bonder, P. Groisman, J.D. Rossi, Numerical approximation of a parabolic problem with a nonlinear boundary condition in several space dimensions, Discrete Cont. Dyn. Systems Ser. B 2 (2002) 279–294.
[6] S.B. Angenent, J.J.L. Velázquez, Asymptotic shape of cusp singularities in curve shortening, Duke Math. J. 77 (1995) 71–110.
[7] C. Bandle, H. Brunner, Blowup in diffusion equations: a survey, J. Comput. Appl. Math. 97 (1998) 3–22.
[8] J. Bebernes, D. Eberly, Mathematical Problems from Combustion Theory, Springer, Berlin, 1989.
[9] Y.-G. Chen, Asymptotic behaviours of blowing-up solutions for finite difference analogue of $u_t=u_{xx}+u^{1+\alpha}$, J. Fac. Sci. Univ. Tokyo Sect. IA, Math. 33 (1986) 541–574.
[10] Y.-G. Chen, Blow-up solutions to a finite difference analogue of $u_t=u_{xx}+u^{1+\alpha}$ in N-dimensional balls, Hokkaido Math. J. 21 (1992) 447–474.
[11] J.R. Dormand, P.J. Prince, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. 6 (1980) 19–26.
[12] R.G. Durán, J.I. Etcheverry, J.D. Rossi, Numerical approximation of a parabolic problem with a nonlinear boundary condition, Discrete Cont. Dyn. Systems 4 (1998) 497–506.
[13] M. Escobedo, M.A. Herrero, Boundedness and blow up for a semilinear reaction–diffusion system, J. Differential Equations 89 (1991) 176–202.

[14] S.O. Fatunla, Numerical Methods for Initial Value Problems in Ordinary Differential Equations, Academic Press, New York, 1988.

[15] J. Fernández Bonder, P. Groisman, J.D. Rossi, On numerical blow-up sets, Proc. Amer. Math. Soc. 130 (2002) 2049–2055.

[16] A. Friedman, B. McLeod, Blow-up of positive solution of semilinear heat equations, Indiana Univ. Math. J. 34 (1985) 425–447.

[17] H. Fujita, On the blowing up of solutions of the Cauchy problem for $u_t = \Delta u + u^{1+\alpha}$, J. Fac. Sci. Univ. Tokyo Sect. IA, Math. 13 (1966) 109–124.

[18] P. Groisman, J.D. Rossi, Asymptotic behaviour for a numerical approximation of a parabolic problem with blowing up solutions, J. Comput. Appl. Math. 135 (2001) 135–155.

[19] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, second rev. ed., Springer, Berlin, 2000 (Fortran source of `DOPRI5` is available at the web-site ⟨http://www.unige.ch/math/folks/hairer/software.html⟩).

[20] T. Ishiwata, S. Yazaki, On the blow-up rate for fast blow-up solutions arising in an anisotropic crystalline motion, J. Comput. Appl. Math. 159 (2003) 55–64.

[21] J.D. Lambert, Computational Methods in Ordinary Differential Equations, Wiley, New York, 1973.

[22] S. Moriguti, C. Okuno, R. Suekane, M. Iri, K. Takeuchi, Ikiteiru Suugaku—Suuri Kougaku no Hatten (in Japanese), Baifukan, Tokyo, 1979.

[23] T. Nakagawa, Blowing up of a finite difference solution to $u_t = u_{xx} + u^2$, Appl. Math. Opt. 2 (1976) 337–350.

[24] F. Quirós, J.D. Rossi, J.L. Vazquez, Complete blow-up and thermal avalanche for heat equations with nonlinear boundary conditions, Comm. Partial Differential Equations 27 (2002) 395–424.

[25] A.A. Samarskii, V.A. Galaktionov, S.P. Kurdyumov, A.P. Mikhailov, Blow-up in Quasilinear Parabolic Equations, Walter de Gruyter, Berlin, 1995.

[26] J.M. Sanz-Serna, J.G. Verwer, A study of the recursion $y_{n+1} = y_n + \tau y_n^m$, J. Math. Anal. Appl. 116 (1986) 456–464.

[27] A. Sidi, Practical Extrapolation Methods—Theory and Applications, Cambridge University Press, Cambridge, 2003.

[28] B. Straughan, Explosive Instabilities in Mechanics, Springer, Berlin, 1998.

[29] J.F. Traub, Iterative Methods for the Solution of Equations, Chelsea Pub. Comp., New York, 1982.

[30] T.K. Ushijima, On the approximation of blow-up time for solutions of nonlinear parabolic equations, Publ. RIMS Inst. Math. Sci. 36 (2000) 613–640.

[31] W. Walter, On existence and nonexistence in the large of solutions of parabolic differential equations with a nonlinear boundary condition, SIAM J. Math. Anal. 6 (1975) 85–90.