On the method of modified equations.
III. Numerical techniques based on the second equivalent equation for the Euler forward difference method

F.R. Villatoro *, J.I. Ramos

Departamento de Lenguajes y Ciencias de la Computación, E.T.S. Ingenieros Industriales, Universidad de Málaga, Plaza El Ejido, s/n, 29013-Málaga, Spain

Abstract

Direct-correction and asymptotic successive-correction methods based on the second equivalent equation are applied to the Euler forward explicit scheme. In direct-correction, the truncation error terms of the second equivalent equation which contain higher-order derivatives together with a starting procedure, are discretized by means of finite differences. Both explicit and implicit direct-correction schemes are presented and their stability regions are studied. The asymptotic successive-correction numerical technique developed in Part II of this series with a consistent starting procedure is applied to the second equivalent equation. Both all-backward and all-centered asymptotic successive-correction methods are presented. The numerical methods introduced in this paper are applied to autonomous and non-autonomous, scalar and systems of ordinary differential equations and compared with the results of second- and fourth-order accurate Runge–Kutta methods. It is shown that the fourth-order Runge–Kutta method is more accurate than the successive-correction techniques for large time steps due to the need for higher-order derivatives of the Euler solution; however, for sufficiently small time steps, but larger enough so that round-off errors are negligible, both methods have nearly the same accuracy. © 1999 Elsevier Science Inc. All rights reserved.

Keywords: Modified equations; Numerical methods; Finite differences; Deferred corrections; Asymptotic successive-correction methods

* Corresponding author. E-mail: villa@lcc.uma.es.

0096-3003/99/$ – see front matter © 1999 Elsevier Science Inc. All rights reserved.
PII: S0096-3003(98)10029-2
1. Introduction

This paper is the third of a series dealing with the assessment of the method of modified equations as a means for both the analysis of finite difference schemes and the development of new numerical ones.

In Part I [1], the technique of modified equations was applied to a non-autonomous ordinary differential equation which was solved numerically by means of the Euler forward scheme. The non-uniqueness of the modified equation was justified, and three representative equations referred to as first, second and third modified equations or equivalent, second equivalent and (simply) modified equations, respectively, were defined. For linear problems, the validity of the method of modified equations was rigorously proved in Part I [1] where an analysis of the first modified or equivalent equation based on asymptotic methods with the step size as a small parameter, was presented. In Part I, it was also shown that, when straightforward or multiple scales asymptotic techniques yield uniform asymptotic expansions, the accuracy of these expansions is a clear indication on the validity of the method of modified equations based on the equivalent equation, and this method was found to be adequate for linear problems. If the asymptotic analysis is based on the equivalent, the second equivalent or the modified equation, the asymptotic results obtained are exactly the same as those of the first one, and, therefore, the asymptotic equivalence among these three different modified equations was claimed in Part I.

In Part II [2], the validity of the modified equation method based on the first modified or equivalent equation was studied as a method for the development of new numerical methods. There, it was shown that the direct numerical correction of the truncation error terms in the equivalent equation yields unstable schemes. A technique for the development of higher-order, successive-correction, stable numerical methods was presented in Part II [2]. This technique is based on a (numerical) asymptotic expansion of the equivalent equation, and asymptotic corrections were applied up to fifth-order for the Euler forward method. In Part II, it was also shown that the Euler equation is nonlinear and that the correction equations are linearized versions of the Euler equation forced with terms which depend on the higher-order derivatives of the solution of the Euler equation. This procedure has good stability properties. In order to start these methods, a new consistent procedure which solves the numerical problem for a number of fictitious points, was introduced. The resulting successive-correction equations were applied to first-order ordinary differential equations and extended to systems of ordinary differential equations. Comparisons with fourth-order accurate Runge–Kutta numerical techniques indicate that the Runge-Kutta methods are more accurate and more efficient than the successive-correction techniques based on the Euler forward scheme. Nonetheless, the numerical results illustrated the
validity of the method of modified equations based on the equivalent equation for the development of new numerical methods with good stability properties.

In this paper, the validity of the modified equation based on the second equivalent equation is studied as a method for the development of new numerical techniques. In Section 2, a direct numerical correction of the truncation error terms in the second equivalent equation is introduced. For the numerical approximations of the higher-order derivatives which appear in the truncation error terms, backward derivatives that result in explicit schemes are presented in Section 2.1 and the stability regions of these higher-order methods are determined. In Section 2.2, centered finite difference approximations for the truncation error terms are used, implicit schemes are obtained, and their stability regions are studied. In Section 2.1, it is shown that the stability of the direct numerical correction to the second equivalent equation is in marked contrast with the instability of the direct numerical correction of the equivalent equation [2].

The asymptotic successive-correction techniques developed in Part II [2] are applied in Section 3 to the correction of the second equivalent equation and used to obtain higher-order numerical methods with good stability properties. In that section, it is shown that, depending on the difference formulas used for the higher-order derivatives in the correction equations, different successive-correction techniques may be obtained. In Sections 3.1 and 3.2, backward and centered formulas, respectively, are used for the truncation error terms. In order to start the resulting methods, the consistent starting procedure introduced in Section 3.3 of Part II [2] which solves the Euler and correction equations backward in time with the same difference formulas used forward in time, is employed.

The direct-correction and successive-correction numerical techniques developed in this paper are applied to systems of ordinary differential equations in Section 4. In Section 4.1, the direct numerical correction with backward evaluations of the derivatives presented in Section 2.1 is extended to systems of ordinary differential equations. In Section 4.2, a new asymptotic successive-correction technique is introduced by using forward formulas for all the higher-order derivatives in the higher-order correction equations, thus avoiding the need for starting procedures.

In Section 5.1, the explicit and implicit direct-correction numerical methods and the all-backward and all-centered, asymptotic successive-correction techniques developed in this paper are compared with fourth-order Runge–Kutta techniques for several nonlinear, first-order, ordinary differential equations. The numerical order, the errors and the complexity of the successive-correction techniques are studied in that section. Some applications of the methods developed in this paper for systems of ordinary differential equations are presented in Section 5.2.
Finally, Section 6 is devoted to the presentation of the main conclusions on both the accuracy of the new numerical techniques presented in this paper and the validity of the method of modified equations.

2. Second equivalent equation for ordinary differential equations

As in Part I [1], we shall consider the simplest first-order, non-autonomous, ordinary differential equation

$$\frac{du(t)}{dt} = F(u(t), t), \quad u(0) = a,$$

where $F$ is a function regular enough so that the problem is well posed (e.g., $F$ is smooth and Lipschitz continuous), and the simplest numerical finite difference scheme for this equation, i.e., the Euler forward scheme,

$$\frac{U_{n+1} - U_n}{k} = F(U^n, t^n), \quad U^0 = a,$$

where $t^n = nk$ and $U^n$ is a numerical approximation to $u(nk)$. Eq. (2) is also here referred to as Euler’s equation or Euler’s method.

The second modified or second equivalent equation for the Euler forward method is the following pseudo-differential equation

$$DU(t) = \frac{kD}{e^{kD} - 1} F(U(t), t), \quad U(0) = a,$$

where $D$ indicates the time differential operator and $U(t)$ is a formal analytical continuation of the discrete sequence $U^n$ so that $U(nk) = U^n$, and $U(t)$ is referred to as correct-value function in Ref. [3]. The Taylor series expansion of the pseudo-differential operator yields the following differential equation,

$$\frac{dU(t)}{dt} = F(U(t), t) - \frac{k}{2} \frac{d}{dt} F(U(t), t) + \frac{k^2}{12} \frac{d^2}{dt^2} F(U(t), t)$$

$$- \frac{k^4}{720} \frac{d^4}{dt^4} F(U(t), t) + O(k^6),$$

which can be considered as the true differential problem solved by the finite difference scheme.

It is important to note that for the well-posedness of Eq. (4), i.e., the second equivalent equation, only one initial condition is necessary, although, if this equation is truncated to any order, further initial conditions for the higher-order derivatives are required [1].

Eq. (4) can be used for the correction of the truncation error terms of Eq. (2), thus allowing for the development of new numerical methods of higher order of accuracy. Each higher-order derivative in the truncation error terms (with opposite sign) can be numerically approximated by a finite difference
formula. After each approximation, the second equivalent equation of the new difference scheme must be used for the next correction. This procedure generates different numerical methods depending on the kind of finite difference formulae used for the higher-order derivatives in the truncation error terms. In the next sections, we shall consider several methods based on different discretizations of the second equivalent equation.

2.1. Explicit direct-correction methods

Consider the method outlined in the last paragraph for the second equivalent equation of the Euler forward method, i.e., Eq. (4), and use backward difference formulas in order to get a sequence of explicit methods. If the truncation error terms of Eq. (4) are truncated to first order in the step size and the sign of the leading-order error is changed, then

\[ \frac{dU(t)}{dt} = F(U(t), t) + \frac{k}{2} \frac{d}{dt} F(U(t), t), \]  

which, by using a backward formula for the derivative of the nonlinearity in the right-hand side of this equation, can be approximated by the following second-order difference scheme,

\[ U_{n+1} - U_n = kF^n + \frac{1}{2} k(F^n - F^{n-1}) = \frac{1}{2} k(3F^n - F^{n-1}), \]

where \( F^n = F(U^n, t^n) \).

In order to start this second-order method, a fictitious time at \( n = -1 \) must be introduced and the solution at this point can be calculated by means of the trapezoidal rule method (applied one step backwards in time), i.e.,

\[ U_{n+1} - U_n = kF^n + \frac{1}{2} k(F^{n+1} - F^n) = \frac{1}{2} k(F^n + F^{n+1}), \]

which can be solved by Newton’s method.

The stability polynomial of Eq. (6) is

\[ \xi^2 - \xi = \frac{1}{2} k(3\xi - 1), \]

which indicates that this scheme is strongly stable and satisfies the root condition.

In order to obtain a third-order numerical method, the second equivalent equation for the Euler forward scheme, i.e., Eq. (4), truncated to \( O(k^2) \), with the first- and second-order derivatives numerically approximated, cannot be used to obtain a consistent numerical approximation because the first-order numerical approximation used in the right-hand side of Eq. (5) for the first-order derivative of the nonlinearity, cf. Eq. (6), introduces new truncation error terms of second order which have to be taken into account. In order to avoid
this problem, a second-order formula may be used in the right-hand side of Eq. (5). In this paper, however, the second equivalent equation of Eq. (6), i.e.,
\[
\frac{dU(t)}{dt} = F(U(t), t) - \frac{5k^2}{12} \frac{d^2}{dt^2} F(U(t), t) + O(k^3),
\]
(9)
is used as the basis for the correction of Eq. (6) in order to obtain a third-order method. Using this second equivalent equation, the differential equation to be solved numerically is
\[
\frac{dU(t)}{dt} = F(U(t), t) + \frac{k}{2} \frac{d}{dt} F(U(t), t) + \frac{5k^2}{12} \frac{d^2}{dt^2} F(U(t), t),
\]
(10)
which, by using first-order backward formulas for the derivatives in its right-hand side, yields the following third-order difference scheme,
\[
U^{n+1} - U^n = kF^n + \frac{1}{2} k(F^n - F^{n-1}) + \frac{5}{12} k(F^n - 2F^{n-1} + F^{n-2})
\]
\[
- \frac{1}{12} k(23F^n - 16F^{n-1} + 5F^{n-2}).
\]
(11)

In order to start this method two fictitious points must be introduced. Here, Eqs. (7) and (6) have been applied at \(n = -2\) and \(n = -1\), respectively, and solved by means of Newton’s method.

The stability polynomial of Eq. (11) is
\[
\xi^3 - \xi^2 = \frac{1}{12} \lambda (23\xi^2 - 16\xi + 5),
\]
(12)
which indicates that this scheme is strongly stable and relatively stable for \(|\lambda|k < 0.545\), and satisfies the root condition.

The procedure presented in previous paragraphs can be used to obtain higher-order numerical methods by successively correcting terms in the second equivalent equations of the numerical methods obtained at each successive stage. For example, the addition of the leading truncation error term corresponding to Eq. (11) to Eq. (10), yields
\[
\frac{dU(t)}{dt} = F + \frac{k}{2} \frac{dF}{dt} + \frac{5k^2}{12} \frac{d^2F}{dt^2} + \frac{3k^3}{8} \frac{d^3F}{dt^3}.
\]
(13)
Using first-order backward formulas for the first-, second- and third-order derivatives in the right-hand side of Eq. (13) (i.e., the formulas used in Eq. (11) and \(F_{itt} \approx (F^n - 3F^{n-1} + 3F^{n-2} - F^{n-3})/k^3\)), the following fourth-order difference scheme is obtained,
\[
U^{n+1} - U^n = \frac{1}{24} k(55F^n - 59F^{n-1} + 37F^{n-2} - 9F^{n-3}).
\]
(14)
In order to start this method, three fictitious points must be introduced; here, Eqs. (7), (6) and (11) have been applied at \(n = -3\), \(-2\) and \(-1\), respectively, and solved by means of Newton’s method. The stability polynomial of Eq. (14)
indicates that this fourth-order method is strongly stable and relatively stable for $|\lambda|k < 0.3$.

As a final application of the methodology discussed in previous paragraphs, we shall consider a fifth-order method for which the differential equation to be solved numerically is

$$
\frac{dU(t)}{dt} = F(U(t), t) + \frac{k}{2} \frac{d}{dt} F(U(t), t) + \frac{5k^2}{12} \frac{d^2}{dt^2} F(U(t), t) + \frac{3k^3}{8} \frac{d^3}{dt^3} F(U(t), t) + \frac{251k^3}{720} \frac{d^4}{dt^4} F(U(t), t).
$$

(15)

Using a first-order backward formula for the derivatives in the right-hand side of this equation ($F_{ttt} \approx (F^n - 4F^{n-1} + 6F^{n-2} - 4F^{n-3} + F^{n-4})/k^4$), the following fifth-order difference scheme results,

$$
U^{n+1} - U^n = \frac{1}{720} k (1901F^n - 2774F^{n-1} + 2616F^{n-2} - 1274F^{n-3} + 251F^{n-4}).
$$

(16)

The starting of this method may be achieved as in previous schemes, i.e., four fictitious points are introduced and Eqs. (7), (6), (11) and (14) are employed at these fictitious points. The stability polynomial of Eq. (16) indicates that this fifth-order method is strongly stable and relatively stable for $|\lambda|k < 0.163$.

Fig. 1 shows the A-stability (star) diagram of all the explicit direct-correction methods presented in this section; these diagrams are symmetric with respect to the real $\lambda$-axis. This figure illustrates the reduction of the absolute stability region as the order of the method is increased, i.e., as higher-order truncation error terms are accounted for in the modified equation.
2.2. Implicit direct-correction methods

The discretization of the truncation error terms in the second equivalent equation of the Euler forward scheme (cf. Eq. (4)) by means of centered formulas results in implicit finite difference schemes and is considered in this section.

The first-order derivative in the right-hand side of Eq. (5) can be numerically approximated by a forward formula as

\[ U^{n+1} - U^n = kF^n + \frac{1}{2} k(F^{n+1} - F^n) = \frac{1}{2} k(F^{n+1} + F^n), \]  

which is the second-order (implicit), strongly and absolutely stable trapezoidal rule.

The second equivalent equation for Eq. (17) is

\[ \frac{dU(t)}{dt} = F(U(t), t) + \frac{k^2}{12} \frac{d^2}{dt^2} F(U(t), t) + O(k^4) \]  

which yields the following differential equation required to obtain a third-order numerical scheme,

\[ \frac{dU(t)}{dt} = F(U(t), t) + \frac{k}{2} \frac{d}{dt} F(U(t), t) - \frac{k^2}{12} \frac{d^2}{dt^2} F(U(t), t). \]  

This equation can be discretized by means of a centered second-order difference formula for the second-order derivative in its right-hand side to obtain the following third-order implicit scheme,

\[ U^{n+1} - U^n = kF^n + \frac{1}{2} k(F^{n+1} - F^n) - \frac{1}{12} k(F^{n+1} - 2F^n + F^{n-1}) \]
\[ = \frac{1}{12} k(5F^{n+1} + 8F^n - F^{n-1}). \]  

In order to start this method, a fictitious time at \( n = -1 \) is introduced and the solution at this time is determined by means of the trapezoidal rule backwards in time, cf. Eq. (17), and the corresponding nonlinear equation can be solved by means of Newton’s method. The stability polynomial of Eq. (20) shows that this scheme is strongly stable, and relatively stable for \( |\lambda|k < 6 \).

The differential equation to be solved numerically in order to obtain a fourth-order method is

\[ \frac{dU(t)}{dt} = F(U(t), t) + \frac{k}{2} \frac{d}{dt} F(U(t), t) - \frac{k^2}{12} \frac{d^2}{dt^2} F(U(t), t) - \frac{k^3}{24} \frac{d^3}{dt^3} F(U(t), t). \]  

By using a first-order backward formula for the third-order derivative in the right-hand side of this equation (cf. \( F_{ttt} \approx (F^{n+1} - 3F^n + 3F^{n-1} - F^{n-2})/k^3 \)), the following fourth-order implicit difference scheme results,
\[ U^{n+1} - U^n = \frac{1}{24} k (9F^{n+1} + 19F^n - 5F^{n-1} + F^{n-2}). \] (22)

In order to start this method, two fictitious points are introduced and Eqs. (17) and (20) are applied backwards in time. Eq. (22) is strongly stable, and relatively stable for \(|\lambda|k < 3\).

The differential equation to be integrated numerically in order to obtain a fifth-order numerical method is

\[
\frac{dU(t)}{dt} = F(U(t), t) + \frac{k}{2} \frac{d}{dt} F(U(t), t) - \frac{k^2}{12} \frac{d^2}{dt^2} F(U(t), t) - \frac{k^3}{24} \frac{d^3}{dt^3} F(U(t), t) - \frac{19k^3}{720} \frac{d^4}{dt^4} F(U(t), t),
\]

(23)

where the fourth-order derivative in its right-hand side can be approximated by a first-order backward formula (cf. \(F_{tttt} \approx (F^{n+1} - 4F^n + 6F^{n-1} - 4F^{n-2} + F^{n-3})/k^4\)) to yield

\[ U^{n+1} - U^n = \frac{1}{720} k (251F^{n+1} + 646F^n - 264F^{n-1} + 106F^{n-2} - 19F^{n-3}). \] (24)

In order to start this method, three fictitious points are introduced and Eqs. (17), (20) and (22) are applied backwards in time. Eq. (24) is strongly stable and relatively stable for \(|\lambda|k < 1.837\).

The A-stability diagrams of the implicit direct-correction methods presented in this section are shown in Fig. 2. As expected, their absolute stability region decreases as the order of the method increases. Furthermore, the A-stability for the fifth-order implicit scheme, i.e., Eq. (24), is very similar to that of the explicit forward Euler scheme.

![Fig. 2. Absolute stability (star) diagrams (only the upper \(\lambda\)-plane is shown) for the characteristic polynomials of the implicit direct-correction methods, i.e., Eq. (2) (solid line), Eq. (20) (dashed line), Eq. (22) (dash-dotted line) and Eq. (24) (dotted line).]
3. Asymptotic successive-correction method for the second equivalent equation

In this section, the asymptotic successive-correction technique developed in Part II [2] for the numerical correction of the higher-order derivatives in the first equivalent equation is applied to the second equivalent equation. This technique yields a sequence of successive-correction problems, where the original finite difference equation is the Euler equation, and a sequence of difference equations is introduced in order to obtain new finite difference schemes of order as high as desired. The new technique can be implemented in different ways depending on the kind of finite difference stencil used for the evaluation of the higher-order derivatives in the second equivalent equation, and two different approaches are introduced in the next sections. In order to start these successive-correction methods, the technique developed in Section 3.3 of Part II [2] which introduces a number of fictitious points to allow for the application of the same difference formulas used in the starting method as those for the inner points, is employed here.

3.1. All-backward asymptotic successive-correction method

As in Part II [2], we shall consider the standard asymptotic expansion of the global error of the Euler method [3], which, for both a differential equation which is sufficiently differentiable and a sufficiently small time step such that the Euler method is stable, can be written as

\[ u(t) = U(t) + kU_1(t) + k^2U_2(t) + \mathcal{O}(k^3), \]  

where \( U(t) \) is the solution to Eq. (4) and \( U_i(t) \) are sufficiently differentiable higher-order corrections to the analytical continuation of the numerical solution (also called correct-value functions [3]) to be determined hereafter. The Euler method, cf. Eq. (2), is the equation to be corrected with the new numerical technique, and the \( U_i \) are the solutions to the successive-correction equations to be introduced below, so that a higher-order numerical solution is obtained as

\[ \hat{u} = U^n + kU_1^n + k^2U_2^n + k^3U_3^n + k^4U_4^n + \mathcal{O}(k^5). \]  

Introducing Eq. (25) into Eq. (1) yields

\[ U_t + kU_{1t} + k^2U_{2t} + k^3U_{3t} + k^4U_{4t} + \mathcal{O}(k^5) \]
\[ = F + F_U(kU_1 + k^2U_2 + k^3U_3 + k^4U_4) \]
\[ + \frac{1}{2} F_{UU}(k^2U_1^2 + 2k^3U_1U_2 + (U_2^2 + 2U_1U_3)k^4) \]
\[ + \frac{1}{3!} F_{UUU}(k^3U_1^3 + 3k^4U_1^2U_2) + \frac{1}{4!} F_{UUUU}(k^4U_1^4) + \mathcal{O}(k^5), \]  

where the subscripts t and U indicate differentiation and \( F \equiv F(U(t), t) \).
Subtraction of (the second equivalent) Eq. (4) from Eq. (27) and equating equal powers of \( k \) yield the following differential equation at leading order,

\[
U_{1t} - F_U U_1 = \frac{1}{2} (F_t + F_U U_t), \quad U_1(0) = 0. \tag{28}
\]

The numerical solution of Eq. (28) yields the first-correction scheme for the Euler equation (2). For the numerical solution of Eq. (28), the Euler forward method is used in order that the resulting finite difference equation be a forced version of the linearization of the Euler scheme, i.e., Eq. (2), so the explicit difference scheme for Eq. (28) reads

\[
\frac{U_{1}^{n+1} - U_{1}^{n}}{k} - F_U U_1^n = \frac{1}{2} \left( F_t^n + F_U \frac{U_{1}^{n+1} - U_{1}^{n}}{k} \right), \tag{29}
\]

where \( F^n \equiv F(U^n, t^n) \) and \( t^n = nk. \) This correction scheme is linearly stable if \( |1 + kF_U(U^n)| \leq 1, \) since the solution of the Euler equation remains bounded, and, therefore, the right-hand side of Eq. (29) is also bounded.

For the development of higher-order correction equations, the second equivalent equation corresponding to Eq. (29), is required. This second equivalent equation (multiplied by \( k \)) is, in pseudo-differential form,

\[
kD U_{1t} = \frac{k^2 D}{e^{kD} - 1} \left( \frac{F_t}{2} + F_U U_1 + \frac{F_U e^{kD} - 1}{2k} U_1 \right), \quad U_1(0) = 0. \tag{30}
\]

Subtracting from Eq. (27) the second equivalent equations for the Euler and the first-correction schemes, i.e., Eq. (4) and the Taylor series of Eq. (30), respectively, yields, to leading order,

\[
U_{2t} - F_U U_2 = F_U \left( \frac{U_{1t}}{2} - \frac{U_{1}^{n}}{12} \right) + F_U \left( \frac{U_t^2}{6} + F_U \left( \frac{U_1^2}{2} + \frac{U_1 U_t}{2} + \frac{U_t^2}{6} \right) \right) + F_U \left( \frac{U_1}{2} + \frac{U_t}{3} \right), \quad U_2(0) = 0. \tag{31}
\]

This equation can be solved numerically by means of the Euler method to yield the second-correction scheme. The second-order derivative in the right-hand side of this equation can be approximated by the following second-order centered formula,

\[
U_{1t}^n = \frac{U_{1}^{n+1} - 2U_{1}^{n} + U_{1}^{n-1}}{k^2} + O(k^2), \tag{32}
\]

and the first-order derivative of the first-order correction \( U_1 \) can be approximated by the first-order forward formula used in Eq. (29).
These approximations result in the following second equivalent equation for the Euler forward method applied to the second-correction equation, i.e., Eq. (31), which, in pseudo-differential form and after multiplication by $k^2$, can be written as

$$k^2 D U_2 = \frac{k^2 D}{e^{kD} - 1} \left( F_U \left( U_2 + \frac{e^{kD} - 1}{2k} U_1 - \frac{e^{kD} - 2 + e^{-kD}}{12k^2} U \right) \right)$$

$$+ \frac{F_{UU}}{6} \left( 3U_1^2 + 3U_1 \frac{e^{kD} - 1}{k} U + \left( \frac{e^{kD} - 1}{k} U \right)^2 \right)$$

$$+ \frac{F_{tt}}{6} + \frac{F_U}{6} \left( 3U_1 + 2 \frac{e^{kD} - 1}{k} U \right), \quad U_2(0) = 0. \tag{33}$$

Following the procedure presented above, i.e., subtracting from Eq. (27) the second equivalent equations for the Euler and the first two corrections schemes, i.e., Eqs. (4), (30) and (33), respectively, the resulting leading-order equation is

$$U_{3t} - F_U U_3 = F_U \left( \frac{U_{2t}}{2} - \frac{U_{1tt}}{12} - \frac{U_{tt}}{24} \right) + F_{UU} \left( \frac{U_2}{2} + \frac{U_{1t}}{3} - \frac{U_{tt}}{24} \right)$$

$$+ F_{UU} \left( U_1 U_2 + \frac{U_{1tt}}{2} + \frac{U_{2tt}}{2} - \frac{U_{1tt}}{12} - \frac{U_{tt}}{24} \right)$$

$$+ \frac{F_{tt}}{24} + F_{Utt} \left( \frac{U_1}{6} + \frac{U_2}{8} \right) + F_{Utt} \left( \frac{U_3}{4} + \frac{U_{1t}}{3} + \frac{U_{tt}}{8} \right)$$

$$+ F_{UUU} \left( \frac{U_1}{6} + \frac{U_1^2}{4} + \frac{U_1^2 U_2}{6} + \frac{U_1^3}{24} \right), \quad U_3(0) = 0. \tag{34}$$

This equation can be solved numerically by means of the Euler method with the first- and second-order derivatives in its right-hand side approximated as those in the second-correction equation, while the third-order derivative may be approximated by

$$U_{ttt} = \frac{U_{n+1} - 3U^n + 3U^{n-1} - U^{n-2}}{k^3} + O(k). \tag{35}$$

Although this approximation may appear to be too coarse, this is not so because the asymptotic technique used here deals correctly with the truncation error terms introduced by this approximation.

Further use of the procedure sketched in the previous paragraphs allows for the development of higher-order correction schemes for the Euler equation, which result in numerical methods of order as high as desired. For example, the next, fourth-order correction is
which can be solved numerically by means of the Euler method using the same finite difference formulas for the higher-order derivatives as those of the previous correction equations, and

$$U_{n+1}^{\text{err}} = \frac{U_{n+1} - 4U_n + 6U_{n-1} - 4U_{n-2} + U_{n-3}}{k^4} + O(k). \quad (37)$$

In order to start the correction schemes given by Eqs. (31), (34) and (36), the technique developed in Section 3.3 of Part II [2] is used in the present section. This technique introduces fictitious times \( n < 0 \) and employs the same finite difference equation at these fictitious points as that used at later times; the number of fictitious points introduced for the evaluation of \( U_i \) depends on the order of the correction method. For example, for an \( M \)th correction, \( M - 1, M - 2, \ldots \), fictitious points are required by the Euler, first-correction, \ldots, techniques. Moreover, the same Euler forward difference scheme is used for all times, although the corresponding difference equation at the fictitious points must be solved backwards in time. For the Euler
equation, the resulting nonlinear difference scheme can be solved by means of Newton’s method; however, the equation to be solved backwards in time is linear and its solution can be determined analytically in an explicit manner.

The consistency of the successive-correction method developed in this section is based on the validity of the second equivalent equation as a formal differential equation which solved exactly the Euler forward difference scheme. The technique developed in this section has two advantages over that of Part II [2] which is based on the (first) equivalent equation. First, the first-correction equation has no starting problem when the second equivalent equation is used. Second, the correction equation for an \( n \)th order approximation needs an approximation for the \( n \)th order derivative when the equivalent equation is used, instead of only an \((n-1)\)th order one when the second equivalent equation is employed. However, the technique presented in this section has a disadvantage over that of Part II [2] since, for the second equivalent equation methods, the derivation of asymptotic successive-correction technique for autonomous equations is not sufficient to obtain a general method for both autonomous and non-autonomous problems as it is for the equivalent equation method. Therefore, the derivation of successive-correction methods and the resulting nonlinear forcing terms in the right-hand side of the linear correction equations are more cumbersome.

As in the technique developed in Part II, if the time step \( k \) is small enough so that the solution of the Euler method is consistent and stable, i.e., convergent and bounded, then the successive-correction equations are also linearly stable. The consistency and stability of the successive-correction equations indicates that, as \( k \) decreases, the asymptotic expansion of these methods is convergent to the solution of the original differential equation. However, since the correction schemes are based on the higher-order derivatives of the Euler scheme, it is necessary that \( k \) be small enough so that these derivatives are evaluated with sufficient accuracy. This indicates that, as the number of corrections is increased, the time step must be decreased, and, as a consequence, the computational cost and the round-off errors of the method increase; thus, there exist an optimum number of corrections and a maximum order of accuracy. Although the determination of this optimum order is outside the scope of this work, it may be the basis for the development of adaptive order methods.

3.2. All-centered asymptotic successive-correction method

In the numerical technique for the construction of successive-correction problems of higher order based on the second equivalent equation presented in Section 3.1, backward difference formulas for the higher-order derivatives in the second, third and fourth correction equations have been used (cf. Eqs. (32), (35) and (37)). However, other finite difference approximations can also be employed for evaluating these derivatives, even if they do require a larger number of mesh
points than those previously used, since the numerical solution of the Euler method can be determined before any correction equation is calculated.

In this section, another successive-correction technique different from that developed in Section 3.1 is presented. In this technique, all the higher-order derivatives in the correction equations are evaluated by means of centered second-order molecules. When this technique is employed, Eqs. (28)–(34) remain the same, but Eq. (35) must be replaced by

\[ U^n_{ii} = \frac{U^{n+2} - 2U^{n+1} + 2U^{n-1} - U^{n-2}}{2k^2} + O(k^2). \]  

Following the procedure presented in the previous section, i.e., subtracting from Eq. (27) the second equivalent equations for the Euler and the first three corrections schemes, cf. Eqs. (3), (30) and (33), respectively, and the corresponding equation for Eq. (34) but where, now, the difference approximation for the third-order derivative is given by Eq. (38), the resulting differential equation at fourth-order reads

\[ U_{4t} - FU_4 = FU \left( \frac{U_{3t}}{2} + \frac{U_{2tt}}{6} - \frac{U_{3tt}}{24} - \frac{U_{4tt}}{180} \right) + FUU \left( \frac{U_{2t}}{6} + \frac{U_{1t}}{2} - \frac{U_{2tt}}{24} - \frac{U_{3tt}}{12} \right) + FUU \left( \frac{U_{1t}}{2} + \frac{U_{2t}}{3} - \frac{U_{3t}}{8} - \frac{U_{4t}}{45} \right) + FUUU \left( \frac{U_{2t}}{2} + \frac{U_{1t}}{3} - \frac{U_{2tt}}{12} - \frac{U_{3tt}}{24} + \frac{U_{4tt}}{80} \right) + FUUU \left( \frac{U_{1t}}{2} + \frac{U_{2t}}{3} - \frac{U_{3t}}{8} - \frac{U_{4t}}{24} - \frac{U_{5t}}{120} \right) + FUUU \left( \frac{U_{1t}}{2} + \frac{U_{2t}}{3} - \frac{U_{3t}}{8} - \frac{U_{4t}}{24} - \frac{U_{5t}}{80} \right), U_4(0) = 0, \]  

which can be integrated numerically by means of the Euler forward method by approximating the fourth-order derivative in its right-hand side by the centered formula

\[ U''^n = \frac{U^{n+2} - 4U^{n+1} + 6U^n - 4U^{n-1} + U^{n-2}}{k^4} + O(k^2). \] (40)

In order to start the correction schemes developed in this section, the technique developed in Section 3.3 of Part II [2] has been used.


The extension of the concept of second equivalent equation presented in Part I [1] for a difference scheme consistent with a system of non-autonomous ordinary differential equations is straightforward. The direct-correction and asymptotic successive-correction methods developed and studied in the previous sections can also be extended to systems of ordinary differential equations. Here, we shall consider both the explicit direct-correction scheme presented in Section 2.1 but with a new starting procedure, and a new successive-correction scheme based on forward difference formulas for the numerical approximation of the higher-order derivatives that appear in the truncation error terms.

4.1. Explicit direct-correction method

Consider the following system of \( N \) differential equations

\[ \frac{dU_i}{dt} = F_i(U(t), t), \quad U_i(0) = a_i, \quad i = 1, 2, \ldots, N, \] (41)

where \( U(t) = (u_1(t), u_2(t), \ldots, u_N(t)) \), and the Euler forward method for each \( U_i(t) \)

\[ \frac{U_{i}^{n+1} - U_{i}^{n}}{k} = F_i(U^n, t^n), \quad U_i^0 = a_i, \] (42)

where \( i = 1, 2, \ldots, N \) and \( U^n = (U^n_1, U^n_2, \ldots, U^n_N) \).

In order to obtain a second-order technique from the Euler forward method, the truncation error terms of the second equivalent equation for the Euler forward finite difference scheme, i.e.,

\[ \frac{dU_i}{dt} = F_i - \frac{k}{2} \frac{dF_i}{dt} + O(k^2), \quad U_i(0) = a_i, \] (43)

must be corrected.

A second-order numerical method can be obtained by direct correction of this second equivalent equation, truncating its right-hand side to first-order in the step size and changing the sign of the corresponding term, i.e.,
\[
\frac{dU_i}{dt} = F_i + \frac{k}{2} \frac{dF_i}{dt}, \quad U_i(0) = a_i. \tag{44}
\]

This equation can be approximated by using a backward difference formula for the first-order derivative of \(F_i\). Such an approximation results in the following second-order method,

\[
U_i^{n+1} - U_i^n = kF_i^n + \frac{1}{2} k (F_i^n - F_i^{n-1}), \quad U_i^0 = a_i, \tag{45}
\]

where \(F_i^n = F_i(U^n, t^n)\).

The second equivalent equation corresponding to Eq. (45) reads as

\[
\frac{dU_i}{dt} = F_i - \frac{5k^2}{12} \frac{d^2 F_i}{dt^2} + O(k^3), \quad U_i(0) = a_i. \tag{46}
\]

This implies that the differential equation that needs to be solved in order to obtain a third-order numerical method is

\[
\frac{dU_i}{dt} = F_i + k \frac{dF_i}{dt} + \frac{5k^2}{12} \frac{d^2 F_i}{dt^2}, \quad U_i(0) = a_i, \tag{47}
\]

which, by using a backward approximation to the second-order derivative of \(F_i\) and changing its sign, yields the following third-order method

\[
U_i^{n+1} - U_i^n = kF_i^n + \frac{1}{2} k (F_i^n - F_i^{n-1}) + \frac{5}{12} k (F_i^n - 2F_i^{n-1} + F_i^{n-2}), \quad U_i^0 = a_i. \tag{48}
\]

Following the same procedure, an explicit, fourth-order method based on the second equivalent equation can be obtained as

\[
U_i^{n+1} - U_i^n = kF_i^n + \frac{1}{2} k (F_i^n - F_i^{n-1}) + \frac{5}{12} k (F_i^n - 2F_i^{n-1} + F_i^{n-2}) + \frac{3}{8} k (F_i^n - 3F_i^{n-1} - 3F_i^{n-2} - F_i^{n-3}), \tag{49}
\]

and, a fifth-order one as

\[
U_i^{n+1} - U_i^n = kF_i^n + \frac{1}{2} k (F_i^n - F_i^{n-1}) + \frac{5}{12} k (F_i^n - 2F_i^{n-1} + F_i^{n-2}) + \frac{3}{8} k (F_i^n - 3F_i^{n-1} - 3F_i^{n-2} - F_i^{n-3}) + \frac{251}{720} k (F_i^n - 4F_i^{n-1} + 6F_i^{n-2} - 4F_i^{n-3} + F_i^{n-4}), \tag{50}
\]

In order to start all the methods presented in this section, the classical Runge–Kutta method of fourth order has been employed.

\[\text{4.2. Forward asymptotic successive-correction method}\]

If the solution of Eq. (41) is sufficiently differentiable and the time step is sufficiently small, the global error of the Euler scheme, cf. Eq. (42), can be expanded in the following asymptotic series,
\[ u_i(t) = U_i(t) + kU_i(t) + k^2 U_{12}(t) + O(k^3), \quad (51) \]

where \( U_i(t) \) is the solution to the second equivalent equation for the Euler forward method, and \( U_{ij}(t) \) are continuous higher-order corrections to each numerical solution.

Introducing Eq. (51) into Eq. (41) yields

\[
\frac{dU_i}{dt} + k \frac{dU_{1i}}{dt} + k^2 \frac{dU_{12}}{dt} + O(k^3) = F_i(U) + \sum_{j=1}^{N} \frac{\partial F_i(U)}{\partial U_j} \left( kU_{j1} + k^2 U_{j2} \right) + \frac{k^2}{2} \left( \sum_{j=1}^{N} \frac{\partial^2 F_i(U)}{\partial U_j^2} U_{j1}^2 + 2 \sum_{j>k=1}^{N} \frac{\partial^2 F_i(U)}{\partial U_j \partial U_k} U_{j1} U_{k1} \right) + O(k^3), \quad (52)
\]

where \( U = (U_1, U_2, \ldots, U_N) \).

In order to obtain the first-correction equation using the asymptotic successive-correction technique developed in Part II [2] when applied to the Euler forward scheme, we subtract from Eq. (52) the second equivalent equation of the Euler forward, cf. Eq. (43), and, then, equate equal powers of \( k \). These result in the following system of ordinary differential equations to leading order,

\[
\frac{dU_{1i}}{dt} = \sum_{j=1}^{N} \frac{\partial F_i}{\partial U_j} \left( U_{j1} + \frac{1}{2} \frac{dU_i}{dt} \right) - \frac{1}{2} \frac{\partial F_i}{\partial t}, \quad U_{1i}(0) = 0, \quad (53)
\]

which is the differential equation for the first-correction scheme. Using the Euler method and approximating the first-order derivative in the right-hand side by means of, for example, the following third-order, forward approximation

\[
\frac{dU_i^n}{dt} \approx \frac{2U_i^{n+1} - 9U_i^{n+2} + 18U_i^{n+1} - 11U_i^n}{6k} + O(k^3), \quad (54)
\]

the resulting finite difference scheme for the first correction is explicit and needs no starting procedure.

Subtraction from Eq. (52) the second equivalent equations corresponding to the Euler and the first correction schemes, cf. Eqs. (42) and (53), respectively, solved numerically by means of the Euler method with Eq. (54) in its right-hand side, yields the following system of differential equations,

\[
U_{12,t} = \sum_{j=1}^{N} F_{i,j} \left( \frac{U_{j2}}{2} + \frac{U_{j,tt}}{6} \right) + \frac{F_{i,tt}}{6} + \sum_{j=1}^{N} F_{i,j} \left( \frac{U_{j1}}{2} + \frac{U_{j,t}}{3} \right)
\]

\[
+ \sum_{j=1}^{N} F_{i,jj} \left( \frac{U_{j1}}{2} + \frac{U_{j1} U_{j,t}}{2} + \frac{(U_{j,t})^2}{6} \right)
\]

\[
+ \sum_{k>j=1}^{N} F_{i,jk} \left( \frac{U_{j1} U_{k1}}{2} + \frac{U_{k1} U_{j,t}}{2} + \frac{U_{j1} U_{k,t}}{2} + \frac{U_{j,t} U_{k,t}}{3} \right), \quad (55)
\]

with \( U_{12}(0) = 0 \), where \( F_{i,t} \) denotes partial derivative with respect to time and \( F_{i,j} \) is the partial derivative with respect to \( U_j \).
The second-correction equation is obtained by numerically solving Eq. (55) by means of the Euler forward scheme where the second-order derivatives is calculated as

$$\frac{d^2 U^n}{dt^2} = U^n_{t,t} \approx \frac{11U^{n+4}_{t} - 56U^{n+3}_{t} + 114U^{n+2}_{t} - 104U^{n+1}_{t} + 35U^n_{t}}{12k^2} + O(k^3),$$

and the first-order derivatives are calculated by using Eq. (54).

Following the same procedure, the fourth-order correction equation is the Euler forward approximation of the following system of differential equations,

$$U_{i3,t} = \frac{F_{i3}}{24} + \sum_{j=1}^{N} F_{ij} \left( \frac{U_{j2}}{2} + \frac{U_{j1,1} + U_{j1,tt}}{6} + \frac{U_{j1,ttt}}{24} \right)$$

$$+ \sum_{j=1}^{N} F_{ij} \left( \frac{U_{j2}^2}{3} + \frac{U_{j1,tt}}{8} \right) + \sum_{j=1}^{N} F_{ij} \left( \frac{U_{j1,1} + U_{j1,tt}}{6} + \frac{U_{j1,ttt}}{8} \right)$$

$$+ \sum_{j,\neq i}^{N} F_{ij,jj} \left( \frac{U_{j1}^2}{4} + \frac{U_{j1,1} + U_{j1,tt}}{3} + \frac{(U_{j1,ttt})^2}{8} \right)$$

$$+ \sum_{j,\neq i}^{N} \frac{F_{ij,jj}}{6} \left( \frac{U_{j1,1}}{6} + \frac{U_{j1}^2}{4} + \frac{U_{j1,tt}}{6} + \frac{(U_{j1,ttt})^2}{24} \right)$$

$$+ \sum_{k,j=1}^{N} F_{ijk} \left( \frac{U_{k1}U_{j1,t} + U_{j1}U_{k2}}{2} + \frac{U_{k1}U_{j1,t}}{2} + \frac{U_{j1}U_{k1,t}}{2} + \frac{U_{k1}U_{j1,tt}}{2} + \frac{U_{j1}U_{k1,tt}}{2} + \frac{U_{k1}U_{j1,ttt}}{2} + \frac{U_{j1}U_{k1,ttt}}{2} \right)$$

$$+ \sum_{k,j>1}^{N} F_{ijk} \left( \frac{U_{j1}^2}{3} + \frac{U_{j1}U_{k1,t}}{3} + \frac{U_{j1}U_{k1,tt}}{6} + \frac{U_{j1}U_{k1,ttt}}{8} \right)$$

$$+ \sum_{k \neq j} F_{ij,kk} \left( \frac{U_{k1}^2}{4} + \frac{U_{k1}U_{j1,t}}{2} + \frac{U_{j1}U_{k1,t}}{2} + \frac{U_{k1}U_{j1,tt}}{2} + \frac{U_{j1}U_{k1,tt}}{2} + \frac{U_{k1}U_{j1,ttt}}{2} + \frac{U_{j1}U_{k1,ttt}}{2} \right)$$

$$+ \sum_{l>k>1} F_{ijkl} \left( \frac{U_{j1}^2}{3} + \frac{U_{j1}U_{k1,t}}{3} + \frac{U_{j1}U_{k1,tt}}{3} + \frac{U_{j1}U_{k1,ttt}}{4} \right)$$

$$+ \sum_{l<k>1} F_{ijkl} \left( \frac{U_{j1}U_{k1,t}}{2} + \frac{U_{j1}U_{k1,tt}}{2} + \frac{U_{j1}U_{k1,ttt}}{2} \right)$$

(57)
with $U_{13}(0) = 0$, where the third-order derivatives in the right-hand side are evaluated as

$$U_{i,ttt}^n \approx \frac{7U_i^{n+5} - 41U_i^{n+4} + 98U_i^{n+3} - 118U_i^{n+2} + 71U_i^{n+1} - 17U_i^n}{4k^3} + O(k^3),$$

and the first- and second-order derivatives also in the right-hand side are calculated by using Eqs. (54) and (56), respectively.

The differential equation for the fifth-order correction has also been calculated but its expression is omitted here. That equation contains fourth-order derivatives in its right-hand side which have been calculated as

$$U_{i,tttt}^n \approx \frac{17U_i^{n+6} - 114U_i^{n+5} + 321U_i^{n+4} - 484U_i^{n+3}}{6k^4} + \frac{411U_i^{n+2} - 186U_i^{n+1} + 35U_i^n}{6k^4} + O(k^3).$$

The first-, second- and third-order derivatives which also appear in the right-hand side of the fifth-order correction equation have been evaluated using Eqs. (54), (56) and (58), respectively.

It is important to note that the use of forward formulas for the higher-order derivatives in the right-hand side of the correction equations developed in this section, yield one-step difference methods which need no starting procedure.

The forward successive-correction technique for systems of ordinary differential equations presented in this section is analogous to that of the centered successive-correction method developed in Section 3.2. Similarly, backward and centered successive-correction techniques for systems of ordinary differential equations can be developed as in Sections 3.1 and 3.2, respectively.

5. Presentation of results

5.1. Ordinary differential equations

The successive-correction methods presented in the previous sections have been used to obtain the numerical solution of the problems P0–P8 presented in Part II [2].

For the sake of convenience, the Euler forward method and the explicit direct-correction methods developed in Section 2.1, i.e., Eqs. (2), (6), (11), (14) and (16) are referred to as E2E0, E2E1, E2E2, E2E3 and E2E4, respectively; the implicit direct-correction methods developed in Section 2.2, i.e., Eqs. (17), (20), (22) and (24), are referred to as E2I1, E2I2, E2I3 and E2I4, respectively; the successive-correction methods with backward approximations to the higher-order derivatives presented in Section 3.1 with second,
third, fourth and fifth order of consistency are referred to as $E_2 B_1, E_2 B_2, E_2 B_3$ and $E_2 B_4$, respectively; and, finally, the successive-correction methods presented in Section 3.2 with centered derivatives and fourth and fifth order of consistency are referred to as $E_2 C_3$ and $E_2 C_4$, respectively. Note that the methods $E_2 I_0, E_2 B_0$ and $E_2 C_0$ coincide with $E_2 E_0, E_2 B_1$ and $E_2 C_1$, and $E_2 B_2$ with $E_2 C_2$.

The results of the successive-correction methods presented in this paper have been compared with those obtained by means of two Runge–Kutta methods: the second-order modified Euler method and the classical fourth-order accurate Runge–Kutta scheme here referred to as EM and RK, respectively.

Table 1 shows the numerical orders calculated as in the Part II [2] for the linear problems $P_0(\cdot \cdot)$ and $P_0(\cdot \cdot \cdot)$. This table indicates that the numerical order approaches, as $k$ decreases, its theoretical (asymptotic) value for all the methods except for $E_2 E_4$ and $E_2 I_4$ which must be of fifth order but are only of fourth order due to the starting procedure used in Sections 3.1 and 3.2. However, the use of a consistent starting procedure for the successive-correction methods preserves the correct asymptotic order of these techniques.

Table 2 shows the maximum and $L^2$-norm absolute errors for the difference between the numerical and the exact solution of the linear problems

<table>
<thead>
<tr>
<th>Numerical order</th>
<th>$P_0(\cdot \cdot)$</th>
<th>$P_0(\cdot \cdot \cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_2 E_0$</td>
<td>1.0193</td>
<td>1.0095</td>
</tr>
<tr>
<td>$E_2 E_1$</td>
<td>2.0271</td>
<td>2.0135</td>
</tr>
<tr>
<td>$E_2 E_2$</td>
<td>3.0779</td>
<td>3.0389</td>
</tr>
<tr>
<td>$E_2 E_3$</td>
<td>4.2300</td>
<td>4.1195</td>
</tr>
<tr>
<td>$E_2 E_4$</td>
<td>4.3476</td>
<td>4.1785</td>
</tr>
<tr>
<td>$E_2 I_1$</td>
<td>2.0005</td>
<td>2.0000</td>
</tr>
<tr>
<td>$E_2 I_2$</td>
<td>3.0252</td>
<td>3.0123</td>
</tr>
<tr>
<td>$E_2 I_3$</td>
<td>4.0578</td>
<td>4.0285</td>
</tr>
<tr>
<td>$E_2 I_4$</td>
<td>4.2723</td>
<td>4.1338</td>
</tr>
<tr>
<td>$E_2 B_1$</td>
<td>2.0597</td>
<td>2.0302</td>
</tr>
<tr>
<td>$E_2 B_2$</td>
<td>3.0620</td>
<td>3.0304</td>
</tr>
<tr>
<td>$E_2 B_3$</td>
<td>4.1063</td>
<td>4.0522</td>
</tr>
<tr>
<td>$E_2 B_4$</td>
<td>5.1540</td>
<td>5.0743</td>
</tr>
<tr>
<td>$E_2 C_3$</td>
<td>4.1053</td>
<td>4.0511</td>
</tr>
<tr>
<td>$E_2 C_4$</td>
<td>5.1488</td>
<td>5.0727</td>
</tr>
<tr>
<td>EM</td>
<td>2.0342</td>
<td>2.0170</td>
</tr>
<tr>
<td>RK</td>
<td>4.0376</td>
<td>4.0188</td>
</tr>
</tbody>
</table>

Table 1

Numerical order for linear problems $P_0(\cdot \cdot)$ and $P_0(\cdot \cdot \cdot)$ based on the maximum absolute error and the time steps 0.0625, 0.03125, 0.01625 and 0.0078125
Table 2
Maximum and $L^2$-norm absolute errors for linear problems P0(+) and P0(−) with $k = 7.8125 \times 10^{-3}$

<table>
<thead>
<tr>
<th></th>
<th>P0(−)</th>
<th>P0(+)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_{\text{max}}$</td>
<td>$E_{L^2}$</td>
</tr>
<tr>
<td>E2E0</td>
<td>$1.4417 \times 10^{-3}$</td>
<td>$2.2148 \times 10^{-2}$</td>
</tr>
<tr>
<td>E2E1</td>
<td>$9.3996 \times 10^{-6}$</td>
<td>$1.4454 \times 10^{-4}$</td>
</tr>
<tr>
<td>E2E2</td>
<td>$6.6672 \times 10^{-8}$</td>
<td>$1.0252 \times 10^{-6}$</td>
</tr>
<tr>
<td>E2E3</td>
<td>$2.0119 \times 10^{-9}$</td>
<td>$1.7210 \times 10^{-8}$</td>
</tr>
<tr>
<td>E2E4</td>
<td>$2.4649 \times 10^{-9}$</td>
<td>$1.4984 \times 10^{-8}$</td>
</tr>
<tr>
<td>E2I1</td>
<td>$1.8711 \times 10^{-6}$</td>
<td>$2.8772 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$7.3397 \times 10^{-9}$</td>
<td>$1.1286 \times 10^{-7}$</td>
</tr>
<tr>
<td>E2I3</td>
<td>$4.1655 \times 10^{-11}$</td>
<td>$6.3960 \times 10^{-10}$</td>
</tr>
<tr>
<td>E2I4</td>
<td>$8.5690 \times 10^{-12}$</td>
<td>$7.0558 \times 10^{-11}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$1.9550 \times 10^{-6}$</td>
<td>$3.8254 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$1.9599 \times 10^{-8}$</td>
<td>$2.9188 \times 10^{-7}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$1.9189 \times 10^{-10}$</td>
<td>$2.4447 \times 10^{-9}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$1.5592 \times 10^{-12}$</td>
<td>$1.8301 \times 10^{-11}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$1.6352 \times 10^{-10}$</td>
<td>$2.0410 \times 10^{-9}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$8.4865 \times 10^{-13}$</td>
<td>$9.0873 \times 10^{-12}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$3.7643 \times 10^{-6}$</td>
<td>$5.7883 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$1.1495 \times 10^{-11}$</td>
<td>$1.7676 \times 10^{-10}$</td>
</tr>
</tbody>
</table>

P0(+) and P0(−). This table and others [4] not shown here indicate that the errors diminish as the order of the numerical methods increases for sufficiently small step sizes, and that the implicit direct-correction and the successive-correction methods of fifth order are more accurate but the fourth-order methods are less accurate than the RK method. The second-order EM method is less accurate than the second-order methods E2I1 and E2B1, but more accurate than E2E1. The all-centered successive-correction methods are most accurate for P0(−) while the implicit direct-correction method are more accurate for P0(+).

A study [4] (not shown here) of the evolution of the errors for the linear problems P0(+) and P0(−) as a function of the time step indicates that, as \( k \) approaches unity, the error of the higher-order methods is larger than those of the first-order ones because the accuracy of the discretization of the higher-order derivatives required in the evaluation of the higher-order methods degrades as \( k \) is increased; this behaviour is not exhibited by Runge–Kutta methods. However, as \( k \) is decreased, the order of the method is the main factor that controls its accuracy, and the fifth-order methods E2I4, E2B4 and E2C4 become more accurate than the fourth-order RK technique, which, in any case, remains more accurate than any of the other fourth-order methods studied in this paper.
The evolution of the errors in time indicates that the reason for the loss of fifth order of the successive-correction methods E2E4 and E2I4 presented in Table 1 is due to the starting procedure, which introduces some oscillations in the solution. These oscillations degrade the accuracy at the fictitious times and, consequently, the numerical order of the fourth- and fifth-order successive-correction methods. The asymptotic successive-correction methods do not show oscillations, their error behaves monotonically, and no deterioration of their numerical order has been observed.

The numerical order for the nonlinear problems P1–P8 is similar to the one presented in Table 1 for linear ones, i.e., the numerical order approaches, as $k$ decreases, its theoretical (asymptotic) value for all the techniques except for the direct-correction methods E2E4 and E2I4 which must be of fifth order but their order is between four and five depending on the nonlinear problem. The reason for this loss of order of the direct-correction methods is due to the inconsistent treatment of the fictitious times used in the starting procedure. Moreover, it has been observed that, when the exact solution of the differential equation approaches asymptotically a stationary point, the behaviour of the error at the initial time is the main factor that controls both the maximum error and the order of the method; therefore, the oscillations at the first steps appear to control the order of the method. When the exact solution grows unboundedly in time, e.g., P1(+), the error grows with time and the oscillations near $t = 0$ do not influence the numerical order significantly.

Tables 3 and 4 show the maximum, $E_{\text{max}}$, and $L^2$, $E_{L2}$, norms of the absolute (in some cases, relative) errors of the difference between the numerical and the exact solutions for some of the nonlinear problems P1–P8. From these tables and others [4] not shown here, it is observed that the errors decrease as the order of the numerical methods is increased. Fig. 3 shows the evolution of the maximum error as $k$ is varied for problem P1(−) for all the methods used in this paper. The results presented in Fig. 3 for P1(−) are representative of the results obtained for the other nonlinear problems P1–P8 which are not presented here.

Fig. 3 shows that, for large $k$ albeit within the stability interval of the methods, the errors of higher-order direct-correction and successive-correction methods are larger than those of the Euler forward method EF0 because the evaluation of the higher-order derivatives in the higher-order correction equations is not sufficiently accurate and is subject to noise; the error of the EF0 method is always larger than that of the Runge–Kutta methods EM and RK. As $k$ is decreased, the numerical evaluation of the lower-order derivatives becomes more accurate, but that of the higher-order ones becomes inaccurate; this implies that E2B3;E2B4 are less accurate than E2B1;E2B2, which, in turn, are more accurate than E2E0. For sufficiently small $k$, the order of each method becomes the main factor that controls the error, and the methods considered in this paper can be ranked for most of the problems considered
Table 3
Maximum and $L^2$-norm absolute errors for P6a, P6b and P6c with $k = 7.8125 \times 10^{-3}$

<table>
<thead>
<tr>
<th></th>
<th>P6a</th>
<th></th>
<th>P6b</th>
<th></th>
<th>P6c</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_{max}$</td>
<td>$E_{L2}$</td>
<td>$E_{max}$</td>
<td>$E_{L2}$</td>
<td>$E_{max}$</td>
<td>$E_{L2}$</td>
</tr>
<tr>
<td>E2E0</td>
<td>$3.6161 \times 10^{-2}$</td>
<td>$1.7657 \times 10^{-2}$</td>
<td>$5.2849 \times 10^{-4}$</td>
<td>$1.7657 \times 10^{-2}$</td>
<td>$7.1893 \times 10^{-4}$</td>
<td>$2.2031 \times 10^{-2}$</td>
</tr>
<tr>
<td>E2E1</td>
<td>$3.4072 \times 10^{-1}$</td>
<td>$8.0864 \times 10^{-2}$</td>
<td>$1.8962 \times 10^{-6}$</td>
<td>$5.1924 \times 10^{-5}$</td>
<td>$1.8962 \times 10^{-6}$</td>
<td>$2.1516 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2E2</td>
<td>$3.0944 \times 10^{-1}$</td>
<td>$8.0781 \times 10^{-2}$</td>
<td>$1.1179 \times 10^{-6}$</td>
<td>$5.5302 \times 10^{-6}$</td>
<td>$1.2812 \times 10^{-8}$</td>
<td>$3.2922 \times 10^{-7}$</td>
</tr>
<tr>
<td>E2E3</td>
<td>$3.5884 \times 10^{-1}$</td>
<td>$8.0878 \times 10^{-2}$</td>
<td>$1.1294 \times 10^{-6}$</td>
<td>$5.9799 \times 10^{-6}$</td>
<td>$5.9581 \times 10^{-10}$</td>
<td>$3.8552 \times 10^{-9}$</td>
</tr>
<tr>
<td>E2E4</td>
<td>$6.4227 \times 10^{-1}$</td>
<td>$8.1182 \times 10^{-2}$</td>
<td>$1.4475 \times 10^{-6}$</td>
<td>$6.1867 \times 10^{-6}$</td>
<td>$6.8382 \times 10^{-10}$</td>
<td>$1.3314 \times 10^{-9}$</td>
</tr>
<tr>
<td>E2I1</td>
<td>$2.6045 \times 10^{-3}$</td>
<td>$8.0740 \times 10^{-2}$</td>
<td>$3.5337 \times 10^{-7}$</td>
<td>$1.1978 \times 10^{-5}$</td>
<td>$3.5321 \times 10^{-7}$</td>
<td>$1.0161 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2I2</td>
<td>$3.0384 \times 10^{-3}$</td>
<td>$8.0708 \times 10^{-2}$</td>
<td>$1.8988 \times 10^{-7}$</td>
<td>$5.8831 \times 10^{-6}$</td>
<td>$1.4135 \times 10^{-9}$</td>
<td>$3.6288 \times 10^{-8}$</td>
</tr>
<tr>
<td>E2I3</td>
<td>$2.8209 \times 10^{-3}$</td>
<td>$8.0712 \times 10^{-2}$</td>
<td>$2.8268 \times 10^{-7}$</td>
<td>$5.8043 \times 10^{-6}$</td>
<td>$7.2771 \times 10^{-11}$</td>
<td>$2.5201 \times 10^{-10}$</td>
</tr>
<tr>
<td>E2I4</td>
<td>$3.3930 \times 10^{-3}$</td>
<td>$8.0712 \times 10^{-2}$</td>
<td>$3.4227 \times 10^{-7}$</td>
<td>$5.8421 \times 10^{-6}$</td>
<td>$5.1717 \times 10^{-11}$</td>
<td>$6.0020 \times 10^{-11}$</td>
</tr>
<tr>
<td>E2B1</td>
<td>$5.3734 \times 10^{-7}$</td>
<td>$1.8458 \times 10^{-5}$</td>
<td>$5.3734 \times 10^{-7}$</td>
<td>$1.8458 \times 10^{-5}$</td>
<td>$5.3734 \times 10^{-7}$</td>
<td>$1.4165 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2B3</td>
<td>$2.1974 \times 10^{-12}$</td>
<td>$7.5207 \times 10^{-11}$</td>
<td>$2.1965 \times 10^{-12}$</td>
<td>$7.3451 \times 10^{-11}$</td>
<td>$2.3550 \times 10^{-12}$</td>
<td>$6.2690 \times 10^{-11}$</td>
</tr>
<tr>
<td>E2B4</td>
<td>$3.2863 \times 10^{-14}$</td>
<td>$1.0947 \times 10^{-12}$</td>
<td>$1.4522 \times 10^{-13}$</td>
<td>$3.1245 \times 10^{-12}$</td>
<td>$4.9738 \times 10^{-14}$</td>
<td>$7.8847 \times 10^{-13}$</td>
</tr>
<tr>
<td>E2C3</td>
<td>$9.4330 \times 10^{-8}$</td>
<td>$2.9242 \times 10^{-6}$</td>
<td>$9.5661 \times 10^{-12}$</td>
<td>$2.9551 \times 10^{-10}$</td>
<td>$1.3487 \times 10^{-12}$</td>
<td>$3.6227 \times 10^{-11}$</td>
</tr>
<tr>
<td>E2C4</td>
<td>$1.1940 \times 10^{-7}$</td>
<td>$3.7014 \times 10^{-6}$</td>
<td>$2.2649 \times 10^{-11}$</td>
<td>$7.0018 \times 10^{-10}$</td>
<td>$1.4877 \times 10^{-14}$</td>
<td>$1.6252 \times 10^{-13}$</td>
</tr>
<tr>
<td>EM</td>
<td>$2.6042 \times 10^{-3}$</td>
<td>$8.0729 \times 10^{-2}$</td>
<td>$7.4102 \times 10^{-10}$</td>
<td>$2.3269 \times 10^{-8}$</td>
<td>$1.3643 \times 10^{-7}$</td>
<td>$3.0921 \times 10^{-6}$</td>
</tr>
<tr>
<td>RK</td>
<td>$8.6806 \times 10^{-4}$</td>
<td>$2.6910 \times 10^{-2}$</td>
<td>$6.4171 \times 10^{-14}$</td>
<td>$1.4148 \times 10^{-12}$</td>
<td>$6.4171 \times 10^{-14}$</td>
<td>$1.5893 \times 10^{-12}$</td>
</tr>
</tbody>
</table>
accurate ones, unless round-off errors dominate the numerical evaluation of the steps, for which the fifth-order methods E2I4, E2B4 and E2C4 are the most correction techniques developed in this paper, except for still smaller time higher-order derivatives. These tables also show that the accuracy of the presented in Tables 3 and 4 and further results [4] not shown here indicate that can be more efficient than the exact solution (numerically calculated by unless the exact solution is known in explicit form, some numerical methods mined by counting the number of floating point operations, and indicates that, Fig. 3 and Tables 3 and 4). As expected, the complexity of a method increases as the order of the method increases. The EM method was found to be more efficient here (from larger to smaller error) as E2E0, E2E1, EM, E2B1, E2I1, E2E2, E2B2, E2I2, E2E3, E2E4, E2B3, E2I3, E2C3,RK, E2B4, E2I4 and E2C4 (cf. Fig. 3 and Tables 3 and 4).

In order to obtain an accurate numerical solution, the time step must be such that $k \ll O(1/\max|F_U(U)|)$. For this value of the time step, the results presented in Tables 3 and 4 and further results [4] not shown here indicate that the accuracy of RK is higher than those of direct-correction and successive-correction techniques developed in this paper, except for still smaller time steps, for which the fifth-order methods E2I4, E2B4 and E2C4 are the most accurate ones, unless round-off errors dominate the numerical evaluation of the higher-order derivatives. These tables also show that the accuracy of the methods E2E, E2I, E2B and E2C depends on the problem, the step size and the order of these techniques, and, perhaps, a more useful comparison might be based on a fixed step size and order for each problem.

The complexity of the numerical methods studied in this paper was determined by counting the number of floating point operations, and indicates that, unless the exact solution is known in explicit form, some numerical methods can be more efficient than the exact solution (numerically calculated by Newton’s method). As expected, the complexity of a method increases as the order of the method increases. The EM method was found to be more efficient

<table>
<thead>
<tr>
<th></th>
<th>(E_{\text{max}})</th>
<th>(E_{L^2})</th>
<th>(E_{\text{max}})</th>
<th>(E_{L^2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2E0</td>
<td>7.7723 \times 10^{-2}</td>
<td>4.3020 \times 10^{-1}</td>
<td>1.8738 \times 10^{-1}</td>
<td>2.7565</td>
</tr>
<tr>
<td>E2E1</td>
<td>1.4051 \times 10^{-3}</td>
<td>6.0862 \times 10^{-3}</td>
<td>1.2494 \times 10^{-4}</td>
<td>2.7156 \times 10^{-3}</td>
</tr>
<tr>
<td>E2E2</td>
<td>4.0755 \times 10^{-5}</td>
<td>1.5032 \times 10^{-4}</td>
<td>3.5546 \times 10^{-6}</td>
<td>4.6985 \times 10^{-5}</td>
</tr>
<tr>
<td>E2E3</td>
<td>1.6103 \times 10^{-6}</td>
<td>5.2795 \times 10^{-6}</td>
<td>4.5321 \times 10^{-8}</td>
<td>9.8573 \times 10^{-7}</td>
</tr>
<tr>
<td>E2E4</td>
<td>8.0098 \times 10^{-8}</td>
<td>2.3939 \times 10^{-7}</td>
<td>1.2446 \times 10^{-9}</td>
<td>1.8206 \times 10^{-8}</td>
</tr>
<tr>
<td>E2I1</td>
<td>2.8662 \times 10^{-4}</td>
<td>1.2368 \times 10^{-3}</td>
<td>2.5055 \times 10^{-5}</td>
<td>5.4236 \times 10^{-4}</td>
</tr>
<tr>
<td>E2I2</td>
<td>4.6695 \times 10^{-6}</td>
<td>1.7160 \times 10^{-5}</td>
<td>3.9842 \times 10^{-7}</td>
<td>5.2645 \times 10^{-6}</td>
</tr>
<tr>
<td>E2I3</td>
<td>1.2706 \times 10^{-7}</td>
<td>4.1493 \times 10^{-7}</td>
<td>3.4334 \times 10^{-9}</td>
<td>7.5274 \times 10^{-8}</td>
</tr>
<tr>
<td>E2I4</td>
<td>4.7963 \times 10^{-9}</td>
<td>1.4277 \times 10^{-8}</td>
<td>7.0346 \times 10^{-11}</td>
<td>1.0261 \times 10^{-9}</td>
</tr>
<tr>
<td>E2B1</td>
<td>4.3079 \times 10^{-4}</td>
<td>1.8637 \times 10^{-3}</td>
<td>3.0594 \times 10^{-3}</td>
<td>3.2634 \times 10^{-2}</td>
</tr>
<tr>
<td>E2B2</td>
<td>3.5103 \times 10^{-6}</td>
<td>1.2889 \times 10^{-5}</td>
<td>3.2578 \times 10^{-5}</td>
<td>3.0039 \times 10^{-4}</td>
</tr>
<tr>
<td>E2B3</td>
<td>4.5612 \times 10^{-8}</td>
<td>1.4867 \times 10^{-7}</td>
<td>2.4540 \times 10^{-7}</td>
<td>2.1752 \times 10^{-6}</td>
</tr>
<tr>
<td>E2B4</td>
<td>7.5382 \times 10^{-10}</td>
<td>2.2394 \times 10^{-9}</td>
<td>1.3408 \times 10^{-9}</td>
<td>1.2166 \times 10^{-8}</td>
</tr>
<tr>
<td>E2C3</td>
<td>4.5793 \times 10^{-8}</td>
<td>1.4921 \times 10^{-7}</td>
<td>2.4435 \times 10^{-7}</td>
<td>2.1678 \times 10^{-6}</td>
</tr>
<tr>
<td>E2C4</td>
<td>5.8803 \times 10^{-10}</td>
<td>1.7438 \times 10^{-9}</td>
<td>1.1815 \times 10^{-9}</td>
<td>1.1015 \times 10^{-8}</td>
</tr>
<tr>
<td>EM</td>
<td>2.8661 \times 10^{-4}</td>
<td>1.2368 \times 10^{-3}</td>
<td>1.7590 \times 10^{-4}</td>
<td>3.6143 \times 10^{-3}</td>
</tr>
<tr>
<td>RK</td>
<td>3.1060 \times 10^{-11}</td>
<td>1.0003 \times 10^{-10}</td>
<td>2.2732 \times 10^{-10}</td>
<td>4.8715 \times 10^{-9}</td>
</tr>
</tbody>
</table>
than the second-order, direct-correction and successive-correction methods, and RK was found to be more efficient than all the methods of order greater than two developed in this paper, except for the E2B methods in some cases, e.g., P3, for which the methods E2B3 and E2B4 are more efficient.

Fig. 4 shows the maximum error in time for the implicit direct-correction methods, the backward successive-correction methods and the Runge–Kutta numerical methods for problems P6a, P6b, and P6c. For problem P6a, whose solution is only continuous, cf. $C^0$, at $t = 2.5$, Fig. 4 (top left) shows that the implicit (and also the explicit, not shown in the figure) direct-correction methods and the Runge–Kutta methods have a jump in the error at $t = 2.5$ with a degradation of and a reduction (to first order) in accuracy. This loss of order is associated with the loss of accuracy in the evaluation of the higher-order derivatives needed by the higher-order methods. However, Fig. 4 (bottom left) shows that the backward successive-correction methods do not show any jump in the error and preserve their correct asymptotic order. The centered successive-correction methods E2C3 and E2C4 also suffer a jump in the error and a decrease in accuracy to second order. Fig. 4
(left) also shows that the backward successive-correction methods are the most accurate techniques used in this paper for problems with a discontinuous nonlinearity.

For problem P6b, Fig. 4 (top center) illustrates that the implicit (also the explicit, not shown in the figure) direct-correction methods of order greater than two have a jump in their error; the order of all these methods drops to second order. The EM method also has a small jump, but its order is not reduced; rather, it increases to third order. The E2E0 and RK methods do not present any jump in the error and their order is the correct one. Fig. 4 (bottom center) indicates that the backward successive-correction methods do not show any jump in the error and preserve their asymptotic order, although RK is the most accurate method. The centered successive-correction methods E2C3 and E2C4, not shown in the figure, have a jump in the error and their accuracy is reduced to third order.
For problem P6c, cf. Fig. 4 (top right), the jump in the error of the implicit direct-correction methods is slightly noticeable and yields a little oscillation, and, as the asymptotic order grows, the jump increases, thus reducing the order of the higher-order methods, i.e., the method E2I3 is of fourth order but E2I4 is only of third order. The explicit direct-correction method exhibits a behaviour very similar to that of the implicit direct-correction methods. The EM and RK methods exhibit their correct order. Fig. 4 (bottom right) shows that the backward successive-correction methods do not show any jump in the error and have their correct asymptotic order. The centered successive-correction methods also have their correct asymptotic order. For time steps smaller that the one used in Fig. 4, the fifth-order successive-correction techniques are more accurate than the RK method.

We next compare the results presented in this section with those presented in Part II [2] for the successive-correction methods based on the first modified or equivalent equation, cf. the backward EF1–EF4, the centered EC2–EC4 and the all-centered AC1–AC4 successive-correction methods. It must be recalled that EF and EC in Ref. [2] used an inconsistent starting procedure which reduces the order of the fifth-order asymptotic methods by one, but AC are consistently started, so the fifth-order AC method preserves its order. This behaviour of the numerical order is the same as the one observed in this paper, thus indicating the great importance of employing consistent starting procedures. The main difference in the behaviours of the numerical order for the AC and the E2B and E2C successive-correction methods has been found for problems P6a and P6b for which only the E2B methods preserve their correct order and have no visible jump in their error.

The errors of the methods presented in this paper and those presented in Part II [2] indicate that the successive-correction techniques and the implicit direct-correction method developed here are more accurate than the successive-correction schemes developed in Part II [2] for all the problems P1–P8. The dependence of the error on the time step and on time for all successive-correction techniques developed in this paper exhibits similar trends to those of the successive-correction techniques presented in Part II [2], except for the small oscillations in the first time steps observed in the EF and AC methods due to inconsistent starting procedures, and the good accuracy of the E2B methods for problem P6.

Since the nonlinear truncation error terms of the second equivalent equation are more complex than those of the equivalent equation, the successive-correction techniques developed in this paper require a larger number of floating point operations than the ones presented in Part II [2]. Note that the explicit direct-correction methods are the most efficient techniques developed in this paper, but their stability regions decrease as the number of successive corrections is increased. Note also that the Runge–Kutta methods are the most efficient techniques presented in this paper. The computational cost for the
implicit direct-correction methods is between those of the explicit direct-correction schemes and the successive-correction techniques.

The derivation of the successive-correction techniques for non-autonomous problems presented in this paper can be substantially simplified for autonomous ones because the derivatives of the nonlinearity in the right-hand side of the second equivalent equation can be simplified, thus reducing the computational cost. Therefore, the successive-correction techniques are more costly for non-autonomous problems than for autonomous ones. The simplification of the successive-correction technique based on the second equivalent equation for autonomous problems cannot be carried out and is not necessary for the asymptotic successive-correction techniques based on the equivalent equation [2].

5.2. Systems of ordinary differential equations

The successive-correction techniques presented in this paper have been applied to the systems S1–S12 of ordinary differential equations considered in Part II [2] and their solutions have been compared with those of the RK method.

Table 5 shows the behaviour of the numerical order for the fifth-order methods E2E4 and E2F4 when applied to problems S1–S12. The numerical order of these methods for problems S1–S12 coincides with its expected value, except that the accuracy of E2E4 is only of second order for problem S4; for this problem, the accuracy of E2E2–E2E4 methods is also of second order. The accuracy of E2F4 methods is consistent with the asymptotic order of these methods for S1–S8 and S12, but it drops about two orders for S9–S11. This drop may be due to both the (forward) stencils used in the calculations and the rapidly varying solutions that these stencils may not account for in a precise manner. The numerical orders of E2E4 and E2F4 for S9–S11 show that the numerical (forward or backward) approximations to the higher-order derivatives of the modified equation play an important role in determining both the accuracy and the order of the method, especially for problems with rapidly varying solutions. The order of E2E1–E2E3 and E2F1–E2F3, not shown in Table 5, approaches asymptotically its correct value for all the methods except for S9–S11 for reasons analogous to the ones stated above.

Tables 6 and 7 show the maximum and $L^2$-norm absolute error for the E2E, E2F and Runge–Kutta methods used in this paper for systems of equations. The methods presented in these tables can be ranked in terms of accuracy (in order of increasing errors) as E2E4, E2F4, RK, E2F3, E2E3, E2F2, E2E2, EM, E2F1, E2E1 and E2E0 for most of the problems and the smallest step size. For problem S4, the errors of E2E2–E2E4 slightly grow instead of decreasing as the time step is decreased, and the order of these methods is only two as shown in
Table 5. For problem S5, the errors of the methods E2F1–E2F4 are very large for the largest step size employed in this paper, and, although, they do decrease as the step size is decreased, their accuracy is lower than expected even for sufficiently small step sizes; this behaviour may be due to the forward approximations used for the higher-order derivatives whose accuracy deteriorates when the solution varies rapidly.

For problem S8, the methods E2E2–E2E4 seem to be of fifth order and are more accurate than RK because its analytical solution is a cubic polynomial whose higher-order derivatives can be evaluated accurately by means of third-order forward differences.

For problems, S9–S11, EM is more accurate than E2F2–E2F4 for the largest times steps, but, for smaller time steps, EM is less accurate than E2F3 and E2F4. It must be pointed out that problems S9–S11 are stiff; therefore, any explicit method would require very small time steps which would have a negative impact on the accuracy of the approximations to higher-order derivatives because of noise due to round-off errors. Furthermore, one rarely uses explicit techniques for stiff systems unless the interest lies in very accurate results in the initial transients or in the evaluation of the rapid oscillations.

From Tables 5–7 and further numerical tests for methods E2E and E2F [4], it can be concluded that the behaviour of the numerical order and the absolute errors of the successive-correction techniques used in this paper, which require no starting procedure and which have been applied to systems of equations, is very similar to the one presented in Section 5.1 for single equations, i.e., the accuracy in the evaluation of the higher-order derivatives in the higher-order correction equations is the main factor that determines the order of the method and controls its global error for a given nonlinearity.
<table>
<thead>
<tr>
<th></th>
<th>$S_1$</th>
<th></th>
<th>$S_2$</th>
<th></th>
<th>$S_3$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_{\text{max}}$</td>
<td>$E_L$</td>
<td>$E_{\text{max}}$</td>
<td>$E_L$</td>
<td>$E_{\text{max}}$</td>
<td>$E_L$</td>
</tr>
<tr>
<td>E2E0</td>
<td>$8.7995 \times 10^{-4}$</td>
<td>$2.4481 \times 10^{-2}$</td>
<td>$8.7095 \times 10^{-4}$</td>
<td>$1.3496 \times 10^{-3}$</td>
<td>$3.0495 \times 10^{3}$</td>
<td>$8.6294 \times 10^{4}$</td>
</tr>
<tr>
<td>E2E1</td>
<td>$5.6872 \times 10^{-6}$</td>
<td>$1.5673 \times 10^{-4}$</td>
<td>$5.6885 \times 10^{-6}$</td>
<td>$8.7043 \times 10^{-5}$</td>
<td>$3.4987 \times 10^{-1}$</td>
<td>$9.8989$</td>
</tr>
<tr>
<td>E2E2</td>
<td>$3.9824 \times 10^{-8}$</td>
<td>$1.0993 \times 10^{-6}$</td>
<td>$4.0226 \times 10^{-8}$</td>
<td>$6.1551 \times 10^{-7}$</td>
<td>$8.8657 \times 10^{-5}$</td>
<td>$2.5084 \times 10^{-3}$</td>
</tr>
<tr>
<td>E2E3</td>
<td>$2.8823 \times 10^{-10}$</td>
<td>$7.9687 \times 10^{-9}$</td>
<td>$2.9401 \times 10^{-10}$</td>
<td>$4.4987 \times 10^{-9}$</td>
<td>$2.9753 \times 10^{-7}$</td>
<td>$8.4180 \times 10^{-6}$</td>
</tr>
<tr>
<td>E2E4</td>
<td>$1.9195 \times 10^{-12}$</td>
<td>$5.4823 \times 10^{-11}$</td>
<td>$1.9769 \times 10^{-12}$</td>
<td>$3.1304 \times 10^{-11}$</td>
<td>$2.7524 \times 10^{-6}$</td>
<td>$7.7873 \times 10^{-5}$</td>
</tr>
<tr>
<td>E2F1</td>
<td>$4.3039 \times 10^{-6}$</td>
<td>$5.3832 \times 10^{-5}$</td>
<td>$4.3054 \times 10^{-6}$</td>
<td>$5.3586 \times 10^{-5}$</td>
<td>$1.9178 \times 10^{1}$</td>
<td>$5.2940 \times 10^{2}$</td>
</tr>
<tr>
<td>E2F2</td>
<td>$2.9275 \times 10^{-8}$</td>
<td>$3.3649 \times 10^{-7}$</td>
<td>$2.9579 \times 10^{-8}$</td>
<td>$3.3831 \times 10^{-7}$</td>
<td>$7.0219 \times 10^{-2}$</td>
<td>$1.8994$</td>
</tr>
<tr>
<td>E2F3</td>
<td>$1.4561 \times 10^{-10}$</td>
<td>$1.5975 \times 10^{-9}$</td>
<td>$1.4862 \times 10^{-10}$</td>
<td>$1.6223 \times 10^{-9}$</td>
<td>$1.9513 \times 10^{-4}$</td>
<td>$5.1888 \times 10^{-3}$</td>
</tr>
<tr>
<td>E2F4</td>
<td>$2.2824 \times 10^{-12}$</td>
<td>$3.5417 \times 10^{-11}$</td>
<td>$2.3521 \times 10^{-12}$</td>
<td>$3.6314 \times 10^{-11}$</td>
<td>$6.0962 \times 10^{-7}$</td>
<td>$1.6286 \times 10^{-5}$</td>
</tr>
<tr>
<td>EM</td>
<td>$2.2955 \times 10^{-6}$</td>
<td>$3.5297 \times 10^{-5}$</td>
<td>$2.2962 \times 10^{-6}$</td>
<td>$3.5135 \times 10^{-5}$</td>
<td>$1.0287$</td>
<td>$2.9106 \times 10^{1}$</td>
</tr>
<tr>
<td>RK</td>
<td>$7.0091 \times 10^{-12}$</td>
<td>$1.0777 \times 10^{-10}$</td>
<td>$7.1511 \times 10^{-12}$</td>
<td>$1.0942 \times 10^{-10}$</td>
<td>$5.0823 \times 10^{-7}$</td>
<td>$1.4379 \times 10^{-5}$</td>
</tr>
</tbody>
</table>
A comparison between the results presented in this paper and those of Part II [2] indicates that, for problems S1–S8 and S12, the methods E2E and E2F of the former are more accurate than the techniques EC of the latter; however, the opposite is true for problems S9–S11. This behaviour may be attributed to the central differences of EC whose stencils employ fewer forward points than E2E and E2F methods of the same order and, as a consequence, the schemes EC may be more accurate for rapidly varying solutions despite the fact these methods employ asymmetric difference formulas in the first steps, whereas E2E techniques employ the fourth-order Runge–Kutta method as starting procedure; the E2F schemes, on the other hand, are self-starting.

### 6. Conclusions

In this paper, the modified equation method has been studied as a means for the development of new numerical techniques for initial-value problems in ordinary differential equations based on the use of the second modified or second equivalent equation for the improvement of the Euler forward method. Both, direct-correction and asymptotic successive-correction methods have been developed. In the direct-correction methods, each higher-order derivative in the truncation error terms is numerically approximated with opposite sign by a finite difference formula. After each correction is obtained, the second equivalent equation of the new difference scheme must be used for the next one, and depending on the kind of finite difference formulas used for the higher-order derivatives in the truncation error terms of the second equivalent equation, several techniques may be developed.
Two different direct-correction numerical techniques have been developed; backward and centered formulas for the higher-order derivatives in the truncation error terms of the second equivalent equation yield explicit and implicit methods, respectively. The stability regions for all the direct-correction numerical methods presented in this paper have been studied. In contrast with the direct-correction of the (first) equivalent equation, which yield unstable methods, the direct-correction of the second equivalent equation yield more stable schemes. This is one of the advantages of the second equivalent equation over the (first) equivalent equation.

An asymptotic successive-correction technique has been applied to the numerical solution of the second equivalent equation, and this technique yields higher-order stable numerical schemes. In this paper, backward and centered formulas have been used with a consistent starting procedure, and result in the so-called all-backward and all-centered asymptotic successive-correction methods based on the Euler forward scheme. The resulting numerical methods are (theoretically) of as high order of consistency as desired and have good linear stability properties.

The direct-correction and successive-correction numerical methods presented in this paper have been applied to autonomous and non-autonomous, first-order, ordinary differential equations and compared with second- and fourth-order Runge–Kutta methods. It has been shown that the fourth-order Runge–Kutta method is more accurate than the successive-correction techniques for large time steps due to the need for higher-order derivatives of the Euler solution in direct-correction and successive-correction techniques. For sufficiently small time steps, the fourth-order direct-correction and successive-correction methods are as accurate as, whereas those of fifth order are more accurate than, the fourth-order Runge–Kutta scheme. The order of all the methods has been checked by using a formula for the numerical order, and the results indicate that the numerical methods based on the second equivalent equation have their expected asymptotic order, except in some cases.

It has been shown that the computational cost of the successive-correction techniques based on the second equivalent equation is larger than those based on the (first) equivalent equation, and this is one of the drawbacks of the successive-correction methods presented here. However, the right-hand side of the correction equations for the successive-correction based on the second equivalent equation can be simplified for autonomous problems, thus slightly reducing the computational cost for such problems.

The methods presented in this paper have also been extended to systems of ordinary differential equations and higher-order differential problems and compared with Runge–Kutta techniques. Second-order autonomous and non-autonomous, linear and non-linear, stiff and non-stiff problems have been studied. The numerical order and absolute errors have been presented, and the
results are similar to those found for scalar, first-order, ordinary differential equations, except in some cases.

It has also been shown that the accuracy of the successive-correction techniques depends on the accuracy of the difference approximation to the higher-order derivatives in the correction equations, the step size and the nonlinearity of the problem, and that, if either the step size is large or the solution varies rapidly, the numerical order of the methods presented here decreases while their absolute errors increase. However, for sufficiently small step sizes, the methods presented here behave accurately and have their correct asymptotic order. The computational cost of these successive-correction methods is larger than those of Runge–Kutta methods of the same order. Therefore, the successive-correction techniques are not competitive with Runge–Kutta methods of the same order.

The main conclusion of this paper is that the results of the numerical methods developed in this paper are a practical indication on the validity of the method of modified equations based on the second modified or second equivalent equation of two-level finite difference methods for initial-value problems in ordinary differential equations.

Acknowledgements

The research reported in this paper was supported by Project PB94-1494 from the D.G.I.C.Y.T. of Spain.

References