Numerical Treatment of IVP ODEs. . .
in a Nutshell

AT Patera, M Yano

October 31, 2014

1 Preamble

Many phenomena and systems in science and engineering may be accurately modeled by systems of IVP (initial value problem) ODEs (ordinary differential equations). As two simple examples, at rather different scales, we cite the dynamics of an automobile suspension and the motion of the planets in our solar system. In this nutshell we describe the essential preparations and ingredients for numerical solution of this large and important class of problems.

In this nutshell,

> We introduce the concept of temporal discretization and the development of finite difference ODE approximation schemes from either a differentiation or integration perspective.
>
> We provide the definitions of truncation error and consistency; absolute stability; and solution error, convergence, and convergence rate (order). We emphasize the connections between consistency, absolute stability, and convergence.
>
> We discuss the notion of stiff and non-stiff equations and the relative advantages of implicit and explicit schemes in these respective contexts.
>
> We derive several particular (illustrative) schemes: Euler Backward; Euler Forward; Crank-Nicolson. We indicate the application of these schemes first to first-order linear scalar IVP ODEs, then to general systems of first-order IVP ODEs.
>
> We recall the framework by which a system of higher order equations, for example a system of coupled oscillators, may be reduced to a system of first-order IVP ODEs.

We provide theoretical justifications as appendices.

Prerequisites: ODEs: first-order and second-order initial value problems; systems of first-order initial value problems. Numerical calculus: differentiation and integration schemes.

2 A Model Problem

2.1 Formulation

We wish to find a function $w(t)$, $0 < t \leq t_f$, such that

$$\frac{dw}{dt} = g(t, w), \quad 0 < t \leq t_f, \tag{1}$$

$$w(0) = w_0 \ , \tag{2}$$

for g a prescribed suitably smooth function of both arguments. The problem (1)-(2) is an initial value problem (IVP) ordinary differential equation (ODE): we must honor the initial condition (2) at time $t = 0$; we must satisfy the *first*-order ODE (1) for each time $t, 0 < t \leq t_f$.

For much of this nutshell we shall consider the particular *model problem*

$$\frac{du}{dt} = \lambda u + f(t), \quad 0 < t < t_f \ , \tag{3}$$

$$u(0) = u_0 \ , \tag{4}$$

for λ a negative scalar real number and $f(t)$ a prescribed function. Note that (3) is a special case of (1): we choose $g(t, w) \equiv \lambda w + f(t)$, and we rename $u \leftarrow w$. (The latter serves to highlight that we consider the special case of (1)-(2) given by (3)-(4).) We also emphasize that the ODE (3) is a *linear* ODE; in general, (1) need not be linear, for example $g(t, w) = \lambda w^3 + f(t)$.

We can motivate our model problem (3)-(4) physically with a simple heat transfer situation. We consider a body at initial temperature $u_0 > 0$ which is then "dunked" or "immersed" into a fluid flow — forced or natural convection — of ambient temperature zero. (More physically, we may view u_0 as the temperature elevation above some non-zero ambient temperature.) We model the heat transfer from the body to the fluid by a heat transfer coefficient, h. We also permit heat generation within the body, $\dot{q}(t)$, due (say) to Joule heating or radiation. If we now assume that the Biot number — the product of h and the diameter of the body in the numerator, the thermal conductivity of the body in the denominator — is small, the temperature of the body will be roughly uniform in space. In this case, the temperature of the body as a function of time, $u(t)$, will be governed by (3)-(4) for $\lambda \equiv -h\,\mathrm{Area}/\rho c\,\mathrm{Vol}$ and $f(t) \equiv \dot{q}(t)/\rho c\,\mathrm{Vol}$, where ρ and c are the density and specific heat, respectively, and Area and Vol are the surface area and volume, respectively, of the body.

Our model problem (3)-(4) shall provide a foundation on which to construct and understand numerical procedures for much more general problems: (1)-(2) for general g, but also *systems* of ordinary differential equations.

2.2 Some Representative Solutions

We shall first study a few important closed-form solutions to (3)-(4) in order to understand the nature of the equation and also to suggest test cases for our numerical approaches. In

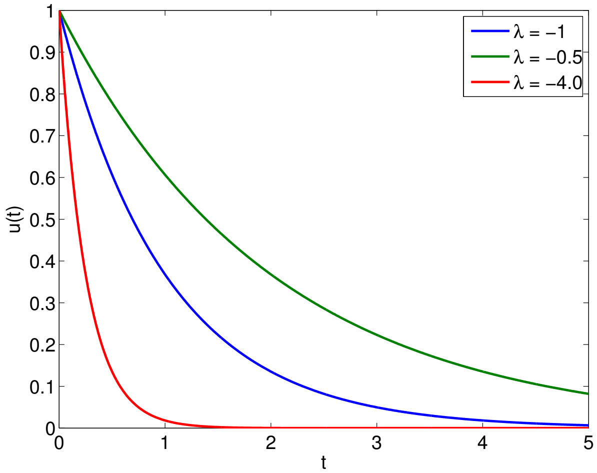


Figure 1: The solution of our model IVP ODE (3)-(4) for $u_0 = 1$, $f(t) = 0$, and several different values of the parameter λ.

all cases these closed-form solutions may be obtained by standard approaches as described in any introductory course on ordinary differential equations.

We first consider the homogeneous case, $f(t) = 0$. We directly obtain

$$u(t) = u_0 e^{\lambda t} \ .$$

The solution starts at u_0 (per the initial condition) and decays to zero as $t \to \infty$ (recall that $\lambda < 0$). The decay rate is controlled by the time constant $1/|\lambda|$ — the larger the λ, the faster the decay. The solutions for a few different values of λ are shown in Figure 1.

Next, we consider the case in which $u_0 = 0$ and $f(t) = 1$. In this case, our solution is given by

$$u(t) = \frac{1}{\lambda}\left(e^{\lambda t} - 1\right) \ .$$

The transient decays on the time scale $1/|\lambda|$ such that as $t \to \infty$ we approach the steady-state value of $-1/\lambda$.

Finally, let us consider a case with $u_0 = 0$ but now a sinusoidal source, $f(t) = \cos(\omega t)$. Our solution is then given by

$$u(t) = \frac{\omega}{\omega^2 + \lambda^2}\sin(\omega t) - \frac{\lambda}{\omega^2 + \lambda^2}\left(\cos(\omega t) - e^{\lambda t}\right) \ . \tag{5}$$

We note that for low frequency there is no phase shift; however, for high frequency there is a $\pi/2$ phase shift. The solutions for $\lambda = -1$, $\omega = 1$ and $\lambda = -20$, $\omega = 1$ are shown in Figure 2. The short-time (initial transient) behavior is controlled by λ and is characterized by the time scale $1/|\lambda|$. The long-time behavior is controlled by the sinusoidal forcing function and is characterized by the time scale $1/\omega$.

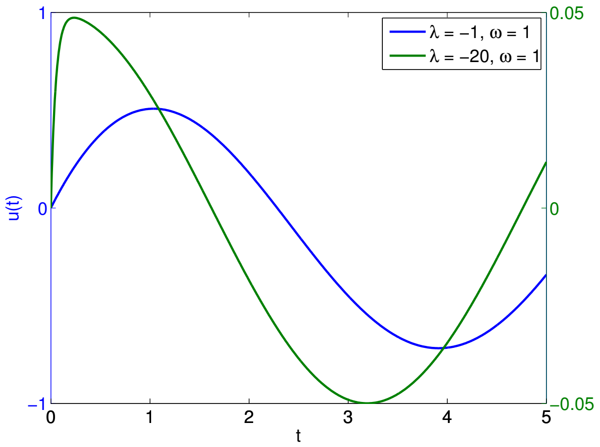


Figure 2: Solutions to our model IVP ODE (3)-(4) for $u_0 = 0$, $f(t) = \cos(\omega t)$, for $\omega = 1$ and two different values of the parameter λ.

3 Euler Backward (EB): Implicit

3.1 Discretization

We first introduce $J+1$ discrete time points, or time levels, given by $t^j = j\Delta t$, $j = 0, 1, \ldots, J$, where $\Delta t = t_f/J$ is the time step. For simplicity, in this nutshell we shall assume that the discrete time levels are equispaced and hence the time step constant; in actual practice, a variable or adaptive Δt can greatly improve computational efficiency. We shall now search for an approximation $\tilde{u}^j$ to $u(t^j)$ for $j = 0, \ldots, J$. We emphasize that our approximation $\tilde{u}^j$ is defined only at the discrete time levels j, $0 \le j \le J$. (Note that $j = 0$ corresponds to $t^0 = 0$, and $j = J$ corresponds to $t^J = t_f$.) We may consider two different approaches to the derivation: differentiation, or integration.

We first consider the differentiation approach. For any given time level t^j, we approximate (i) the time derivative in (3) — the left-hand side of (3) — at time t^j by the first-order Backward Difference Formula,

$$\frac{du}{dt}(t^j) \approx \frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} \ , \tag{6}$$

and (ii) the right-hand side of (3) at t^j by

$$\lambda u(t^j) + f(t) \approx \lambda \tilde{u}^j + f(t^j) \ .$$

We now equate our approximations for the left-hand and right-hand sides of (3) to arrive at

$$\frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} = \lambda \tilde{u}^j + f(t^j) \ . \tag{7}$$

We can apply our difference approximation (6) for $j = 1, \ldots, J$, but not for $j = 0$, since $\tilde{u}^{-1}$ does not exist. Fortunately, we have not yet invoked our initial condition, which provides the final equation: $\tilde{u}^0 = u_0$. Our scheme is thus

$$\frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} = \lambda \tilde{u}^j + f(t^j), \quad j = 1, \ldots, J \; , \tag{8}$$

$$\tilde{u}^0 = u_0 \; . \tag{9}$$

Note that (8)-(9) constitutes $J + 1$ equations in $J + 1$ unknowns. The Euler Backward scheme is *implicit* because the solution at time level j appears on the right-hand side (more precisely, in the approximation of the right-hand side of our differential equation, (3)).

We now consider the integration approach to the derivation of Euler Backward. Let us say that we have in hand our approximation to $u(t^{j-1})$, $\tilde{u}^{j-1}$. We may then write

$$\begin{aligned} u(t^j) &= u(t^{j-1}) + \int_{t^{j-1}}^{t^j} \frac{du}{dt}\, dt \\ &= u(t^{j-1}) + \int_{t^{j-1}}^{t^j} (\, \lambda u + f(t)\,)\, dt \\ &\approx \tilde{u}^{j-1} + \int_{t^{j-1}}^{t^j} (\, \lambda u(t) + f(t)\,)\, dt \\ &\approx \tilde{u}^{j-1} + (\, \lambda \tilde{u}^j + f(t^j)\,)\Delta t \equiv \tilde{u}^j \; , \end{aligned} \tag{10}$$

where in the last step, (10), we apply the *rectangle, right* rule of integration over the segment (t^{j-1}, t^j). We now supplement (10) with our initial condition to arrive at

$$\tilde{u}^j = \tilde{u}^{j-1} + (\, \lambda \tilde{u}^j + f(t^j)\,)\Delta t \, , \quad j = 1, \ldots, J \; , \tag{11}$$

$$\tilde{u}^0 = u_0 \; , \tag{12}$$

which is clearly equivalent to (8)-(9). We note that although the rectangle, right rule is a convenient fashion by which to derive the Euler Backward scheme, there is an important distinction between integration of a known function and integration of an ODE: in the latter we must introduce the additional approximations in red in (10), which in turn admit the possibility of instability; we shall discuss the latter in more depth shortly.

Finally, we can derive, from either (8)-(9) or (11)-(12), a formula by which we can "time step" the Euler Backward approximation forward in time. In particular, we may solve for $\tilde{u}^j$ in (11) (or (8)) to obtain

$$\tilde{u}^j = \frac{\tilde{u}^{j-1} + f(t^j)\Delta t}{1 - \lambda \Delta t}, \quad j = 1, \ldots, J \; , \tag{13}$$

$$\tilde{u}^0 = u_0 \; . \tag{14}$$

We make two remarks. First, we may *march* the solution forward in time: there is no influence of times $t > t^j$ on $\tilde{u}^j$, just as we would expect for an *initial* value problem. We start with $\tilde{u}^0 = u_0$; we may then find $\tilde{u}^1$ in terms of $\tilde{u}^0$, $\tilde{u}^2$ in terms of $\tilde{u}^1$, $\tilde{u}^3$ in terms of

$\tilde{u}^2$, ..., and finally $\tilde{u}^J$ in terms of $\tilde{u}^{J-1}$. Second, at each time level t^j, in order to obtain $\tilde{u}^j$, we must *divide* by $(1 - \lambda\Delta t)$. We shall later consider systems of ODEs, in which case this division by a scalar will be replaced by "division" by a matrix: the latter of course refers to the solution of a system of linear equations, which can be an expensive proposition.

CYAWTP 1. Consider the Euler Backward scheme for our model problem for $u_0 = 1$, $f(t) = 0$, and $\lambda = -2$ for $t_f = 1$. Find $\tilde{u}^J$ for $J = 1$, $J = 2$, $J = 4$, $J = 8$, and $J = 16$. How does $\tilde{u}^J$ compare to the exact solution, $\exp(-2)$, as you increase J (and hence decrease Δt)? What convergence rate $\tilde{u}^J \to \exp(-2)$ might you expect based on your knowledge of the rectangle, right rule of integration?

We anticipate that the solution $\tilde{u}^j$, $j = 1, \ldots, J$, will converge to the true solution $u(t^j)$, $j = 1, \ldots, J$, as $\Delta t \to 0$ such that our finite difference approximation of (6) approaches du/dt. We now summarize the convergence analysis.

3.2 Consistency

We first introduce the *truncation error*: we substitute the solution $u(t)$ of the ODE, (3), into the Euler Backward discretization of the ODE, (8), to define

$$\tau^j_{\text{trunc}} \equiv \frac{u(t^j) - u(t^{j-1})}{\Delta t} - \lambda u(t^j) - f(t^j), \quad j = 1, \ldots, J\ .$$

Note that if our Backward Finite Difference formula exactly reproduced the first derivative then the truncation error would be zero: the truncation error is thus a measure of how well our finite difference formula represents the continuous equation. We are particularly interested in the largest of the truncation errors, defined as

$$\tau^{\max}_{\text{trunc}} \equiv \max_{j=1,\ldots,J} |\tau^j_{\text{trunc}}|\ .$$

The finite difference scheme is *consistent* with the ODE if and only if

$$\tau^{\max}_{\text{trunc}} \to 0 \quad \text{as} \quad \Delta t \to 0\ .$$

As we shall see, consistency is a necessary condition for convergence: the difference equation must well approximate the differential equation.

We now consider our Euler Backward finite difference approximation, (8), to the ODE (3). For f suitably smooth in time, we can derive

$$\tau^{\max}_{\text{trunc}} \le \frac{\Delta t}{2} \max_{t\in[0,t_f]} \left| \frac{d^2u}{dt^2}(t) \right|, \tag{15}$$

as demonstrated in Appendix 8.1. Since $\tau^{\max}_{\text{trunc}} \to 0$ as $\Delta t \to 0$, we conclude that our Euler Backward approximation (8) is consistent with the ODE (3).

3.3 Stability

To study stability, let us consider the homogeneous version of our IVP ODE,

$$\frac{du}{dt} = \lambda u\,, \quad 0 < t \leq t_f\ , \tag{16}$$

$$u(0) = 1\ . \tag{17}$$

Recall that in this case, in which $f(t) = 0$, the exact solution is given by $u(t) = e^{\lambda t}$: $u(t)$ decays as t increases (since $\lambda < 0$). (Note that our choice $u_0 = 1$ for the initial condition is not important in the context of stability analysis.)

We now apply the Euler Backward scheme to this equation to obtain

$$\frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} = \lambda \tilde{u}^j, \quad j = 1, \ldots, J\ , \tag{18}$$

$$u^0 = 1\ .$$

A scheme is *absolutely stable* if and only if

$$|\tilde{u}^j| \leq |\tilde{u}^{j-1}|, \quad j = 1, \ldots, J\ . \tag{19}$$

We note that our absolute stability requirement, (19), is quite natural. We know that the exact solution to (16), $u(t)$, decays in time. We may thus rightfully insist that the numerical approximation to $u(t)$, $\tilde{u}^j$ of (18), should also decay in time.

We pause for three subtleties. First, absolute stability is an intuitive but also a rather strong definition of stability. Although the notion of absolute stability suffices for our purposes here, a weaker definition of stability is required to treat more general equations. Second, in principle, stability depends only on our difference equation, and not on our differential equation. In practice, since we also wish to ensure consistency, the form of the difference equation reflects the differential equation of interest. Third, although it may appear that we have ignored the inhomogeneity, $f(t)$, in fact stability only depends on the homogeneous operator. This conclusion naturally emerges from the error analysis, as we discuss further below.

Let us now demonstrate that the Euler Backward scheme, (18), is absolutely stable for all Δt (under our hypothesis $\lambda < 0$). We first rearrange our difference equation, (18), to obtain $\tilde{u}^j - \tilde{u}^{j-1} = \lambda \Delta t\, \tilde{u}^j$, and hence $\tilde{u}^j(1 - \lambda \Delta t) = \tilde{u}^{j-1}$, and finally $|\tilde{u}^j|\,|1 - \lambda\Delta t| = |\tilde{u}^{j-1}|$. We may then form

$$\frac{|\tilde{u}^j|}{|\tilde{u}^{j-1}|} = \frac{1}{|1 - \lambda \Delta t|} \equiv \gamma\ , \quad j = 1 \ldots, J\ , \tag{20}$$

where γ (here independent of j) is denoted the *amplification factor*. It follows from the definition of γ, (20), that we may rephrase our requirement (19) in terms of the amplfication factor: our scheme is absolutely stable if and only if $\gamma \leq 1$. We now recall that $\lambda < 0$, and we thus directly obtain

$$\gamma = \frac{1}{1 - \lambda \Delta t} < 1 \quad \text{for all} \quad \Delta t > 0\ ; \tag{21}$$

we may thus conclude that Euler Backward discretization of our model problem is absolutely stable for all Δt.

We close with a more refined characterization of absolute stability. A scheme is *unconditionally absolutely stable* if it is absolutely stable for all (positive) Δt. A scheme is *conditionally absolutely stable* if it is absolutely stable only for $\Delta t \leq \Delta t_{\mathrm{cr}}$, where Δt_{cr} is the *critical time step*. (Hence, for an unconditionally absolutely stable scheme, effectively $\Delta t_{\mathrm{cr}} = \infty$.) We observe that, under our assumption $\lambda < 0$, the Euler Backward scheme for our particular model problem, (18), is unconditionally absolutely stable.

3.4 Convergence

A scheme is convergent if the numerical approximation approaches the exact solution as the time step, Δt, is reduced. More precisely, convergence is defined as

$$\tilde{u}^j \to u(t^j) \quad \text{for fixed } t^j = j\Delta t \text{ as } \Delta t \to 0 \ . \tag{22}$$

Note that fixed time $t^j = j\Delta t$ implies that the time level j must tend to infinity as $\Delta t \to 0$. (We note that the perhaps more obvious definition of convergence, $\tilde{u}^j \to u(t^j)$ for fixed j as $\Delta t \to 0$, is, in fact, meaningless: for fixed j, $t^j = j\Delta t \to 0$ as $\Delta t \to 0$, and hence we only confirm convergence to the initial condition.) Note (22) implies that as $\Delta t \to 0$ we must take an infinite number of time steps to arrive at our discrete approximation.

The precise relationship between consistency, stability, and convergence is summarized in the Dahlquist equivalence theorem. The theorem, or more precisely a more restrictive form of this theorem, states that consistency and absolute stability imply convergence. Thus, we need only demonstrate that a scheme is both consistent and stable in order to confirm that the scheme is convergent. Consistency is required to ensure that the truncation error — the error between the difference equation and the differential equation — is small; stability is required to ensure that the truncation errors are controlled — amplification factor $\gamma \leq 1$ — by the difference equation.

In our current context, we may conclude that the Euler Backward approximation, $\tilde{u}^j$, $0 \leq j \leq J$, (8)-(9), converges to the exact solution of our model problem, (3)-(4): consistency, (15), and stability, (21), imply convergence, (22). We explicitly prove convergence, and hence the Dahlquist theorem in this particular case, in Appendix 8.2.

3.5 Convergence Rate: Order

The Dahlquist equivalence theorem assures us that if a scheme is consistent and (absolutely) stable then the scheme is convergent. However, the theorem does not state how rapidly the approximation will converge to the exact solution as the time step, Δt, is reduced. Formally, a scheme is p^{th}-order accurate if

$$|e^j| \leq C\Delta t^p \quad \text{for fixed } t^j = j\Delta t \text{ as } \Delta t \to 0 \ ,$$

where $e^j \equiv u(t^j) - \tilde{u}^j$ is the *solution error.* Note the distinction between the truncation error and the solution error: the former measures the error in the difference equation, whereas the latter measures the error in the approximate solution; the two errors are related through stability.

In general, for a stable scheme, if the truncation error is p^{th}-order accurate, then the scheme is p^{th}-order accurate:

$$\tau_{\text{trunc}}^{\max} \leq C\Delta t^p \quad \Rightarrow \quad |e^j| \leq C\Delta t^p \quad \text{for a fixed } t^j = j\Delta t \ .$$

In other words, once we confirm the stability of a scheme, then we need only analyze the truncation error to predict the convergence rate of the solution error. In general it is relatively simple to estimate, or intuit, the dependence of the truncation error on Δt.

We now consider Euler Backward approximation of our model problem. We may conclude from (15) and (21) that the solution error will decrease as Δt as $\Delta t \to 0$: the Euler Backward scheme is first-order accurate, $p = 1$ (for f suitably smooth). We develop a precise bound in Appendix 8.2: for any positive Δt,

$$|e^j| \leq C\Delta t \quad \text{for fixed } t^j = j\Delta t \ ,$$

where

$$C \equiv \frac{t_f}{2} \max_{t\in[0,t_f]} \left| \frac{d^2u}{dt^2}(t) \right| .$$

Figure 3 presents the convergence behavior of the Euler Backward approximation of our model problem for $u_0 = 1$, $f(t) = 0$, and $\lambda = -4$ for $t_f = 1$. We choose for our time of interest — the fixed time at which we measure the error — the final time, $t = t_f = 1$. We observe that, as predicted by the theory, the solution error decreases as Δt: the slope of the log of the error *vs* the log of Δt is 1.

4 Euler Forward (EF): Explicit

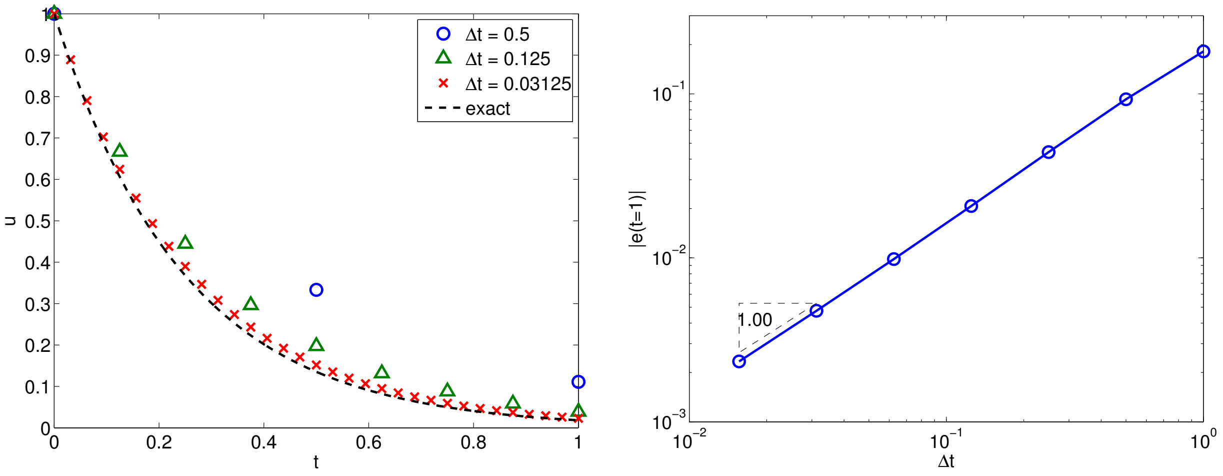
We again introduce $J + 1$ discrete time points, or time levels, given by $t^j = j\Delta t$, $j = 0, 1, \ldots, J$, where $\Delta t = t_f/J$ is the time step. We search for an approximation $\tilde{u}^j$ to $u(t^j)$ for $j = 0, \ldots, J$. (In principle, we should denote our Euler Backward approximation by (say) $\tilde{u}^j_{\text{EB}}$ and our Euler Forward approximation by (say) $\tilde{u}^j_{\text{EF}}$, however more simply we shall rely on context. In this section, $\tilde{u}^j$ shall always refer to Euler Forward.) We may consider two different approaches to the derivation of the Euler Forward scheme: differentiation, or integration.

We first consider the differentiation approach. For any given time level $t^{j'}$, we approximate (i) the time derivative in (3) — the left-hand side of (3) — at time $t^{j'}$ by the first-order Forward Difference Formula,

$$\frac{du}{dt}(t^{j'}) \approx \frac{\tilde{u}^{j'+1} - \tilde{u}^{j'}}{\Delta t} \ , \tag{23}$$

and (ii) the right-hand side of (3) at $t^{j'}$ by

$$\lambda u(t^{j'}) + f(t^{j'}) \approx \lambda \tilde{u}^{j'} + f(t^{j'}) \ .$$



(a) solution

(b) error

Figure 3: The convergence behavior of the Euler Backward approximation of our model problem for $u_0 = 1, f(t) = 0$, and $\lambda = -4$ for $t_f = 1$. Note $|e(t = 1)| \equiv |u(t^j) - \tilde{u}^j|$ at $t^j = j\Delta t = 1$.

Note the one, and only, difference between Euler Forward and Euler Backward: in the former we consider a forward difference, whereas in the latter we consider a backward difference. We now equate our approximations for the left-hand and right-hand sides of (3) to arrive at

$$\frac{\tilde{u}^{j'+1} - \tilde{u}^{j'}}{\Delta t} = \lambda \tilde{u}^{j'} + f(t^{j'}) \ .$$

We next apply our initial condition to arrive at $J + 1$ equations in $J + 1$ unknowns:

$$\begin{aligned} \frac{\tilde{u}^{j'+1} - \tilde{u}^{j'}}{\Delta t} &= \lambda \tilde{u}^{j'} + f(t^{j'}) \ , \quad j = 0, \ldots, J - 1 \ , \\ \tilde{u}^0 &= u_0 \ . \end{aligned} \tag{24}$$

Finally, we shall shift our indices, $j = j' + 1$, such that — for uniform comparison between various schemes — the largest time level in our difference equation is t^j:

$$\frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} = \lambda \tilde{u}^{j-1} + f(t^{j-1}), \quad j = 1, \ldots, J \ , \tag{25}$$

$$\tilde{u}^0 = u_0 \ . \tag{26}$$

The Euler Forward scheme is *explicit* because the solution at time level j does *not* appear on the right-hand side of our difference equation (more precisely, in the approximation of the right-hand side of our differential equation, (3)).

We now consider the integration approach to the derivation of Euler Forward. Let us say that we have in hand our approximation to $u(t^{j-1})$, $\tilde{u}^{j-1}$. We may then write

$$\begin{aligned} u(t^j) &= u(t^{j-1}) + \int_{t^{j-1}}^{t^j} \frac{du}{dt}\,dt \\ &= u(t^{j-1}) + \int_{t^{j-1}}^{t^j} (\,\lambda u + f(t)\,)\,dt \\ &\approx \tilde{u}^{j-1} + \int_{t^{j-1}}^{t^j} (\,\lambda u(t) + f(t)\,)\,dt \\ &\approx \tilde{u}^{j-1} + (\,\lambda \tilde{u}^{j-1} + f(t^{j-1})\,)\Delta t \equiv \tilde{u}^j \;, \end{aligned} \tag{27}$$

where in the last step, (27), we apply the *rectangle, left* rule of integration over the segment (t^{j-1}, t^j). We now supplement (27) with our initial condition to arrive at

$$\tilde{u}^j = \tilde{u}^{j-1} + (\,\lambda\tilde{u}^{j-1} + f(t^{j-1})\,)\Delta t\,, \quad j = 1,\ldots,J\;, \tag{28}$$

$$\tilde{u}^0 = u_0\;, \tag{29}$$

which we see is equivalent to (25)-(26). Note the one, and only, difference between Euler Forward and Euler Backward: in the former we consider rectangle, left, whereas in the latter we consider rectangle, right. We again note, now in the case of Euler Forward, that there is an important distinction between integration of a known function and integration of an ODE: in the latter we must introduce the additional approximations in red in (27), which in turn admit the possibility of instability — in the case of Euler Forward, a very real possibility.

We make two remarks. First, we observe directly from (28)-(29) that we may *march* the solution forward in time: there is no influence of times $t > t^j$ on $\tilde{u}^j$, just as we would expect for an *initial* value problem. We start with $\tilde{u}^0 = u_0$; we may then find $\tilde{u}^1$ in terms of $\tilde{u}^0$, $\tilde{u}^2$ in terms of $\tilde{u}^1$, $\tilde{u}^3$ in terms of $\tilde{u}^2$, ..., and finally $\tilde{u}^J$ in terms of $\tilde{u}^{J-1}$. Second, at each time level t^j, in order to obtain $\tilde{u}^j$, we must *multiply* $\tilde{u}^{j-1}$ by $(1 + \lambda\Delta t)$ (and also multiply $f(t^{j-1})$ by Δt). We shall later consider systems of ODEs, in which case this multiplication by a scalar will be replaced by multiplication by a matrix, a not-so-expensive proposition.

CYAWTP 2. Consider the Euler Forward scheme for our model problem for $u_0 = 1, f(t) = 0$, and $\lambda = -2$ for $t_f = 1$. Find $\tilde{u}^J$ for $J = 1$, $J = 2$, $J = 4$, $J = 8$, and $J = 16$. How does $\tilde{u}^J$ compare to the exact solution, $\exp(-2)$, as you increase J (and hence decrease Δt)? What convergence rate $\tilde{u}^J \rightarrow \exp(-2)$ might you expect based on your knowledge of the rectangle, left rule of integration?

We now analyze the consistency and stability of the scheme. We first consider consistency. It is readily demonstrated, by arguments quite similar to the development provided in 8.1 for Euler Backward, that for the Euler Forward scheme

$$\tau_{\text{trunc}}^{\max} \leq \frac{\Delta t}{2} \max_{t\in(0,t_f)} \left| \frac{d^2 u}{dt^2}(t) \right| \;. \tag{30}$$

We observe that $\tau_{\text{trunc}}^{\max} \to 0$ as $\Delta t \to 0$, and hence the Euler Forward scheme is *consistent* with our ODE (3). We furthemore note that our bound for the truncation error for Euler Forward, (30), is identical to our bound for the truncation error for Euler Backward, (15), and we might thus conclude that the two schemes will in general produce similar results. This is not always the case: we must also consider stability.

To analyze the stability of the scheme we consider the homogeneous ODE. In particular, Euler Forward approximation of (16) yields

$$\frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} = \lambda \tilde{u}^{j-1}, \quad j = 1, \ldots, J \ , \tag{31}$$

$$u^0 = 1 \ .$$

We first rearrange our difference equation, (31), to obtain $\tilde{u}^j - \tilde{u}^{j-1} = \lambda \Delta t \tilde{u}^{j-1}$, hence $|\tilde{u}^j| = |1 + \lambda \Delta t||\tilde{u}^{j-1}|$. We may then form

$$\frac{|\tilde{u}^j|}{|\tilde{u}^{j-1}|} = |1 + \lambda \Delta t| \equiv \gamma \ , \quad j = 1 \ldots, J \ , \tag{32}$$

where γ (here independent of j) is our *amplification factor*. We recall that our condition (19) can be recast in terms of the amplification factor: our Euler Forward scheme is absolutely stable if and only if $\gamma \leq 1$. It follows from (32) that our stability condition is $|1 + \lambda \Delta t| \leq 1$, which we may unfold as

$$-1 \leq 1 + \lambda \Delta t \leq 1 \ , \ \text{or}$$

$$-2 \leq \lambda \Delta t \leq 0 \ .$$

The right inequality provides no information, since $\lambda \Delta t$ is perforce negative due to our assumption $\lambda < 0$ (and $\Delta t > 0$). The left inequality yields

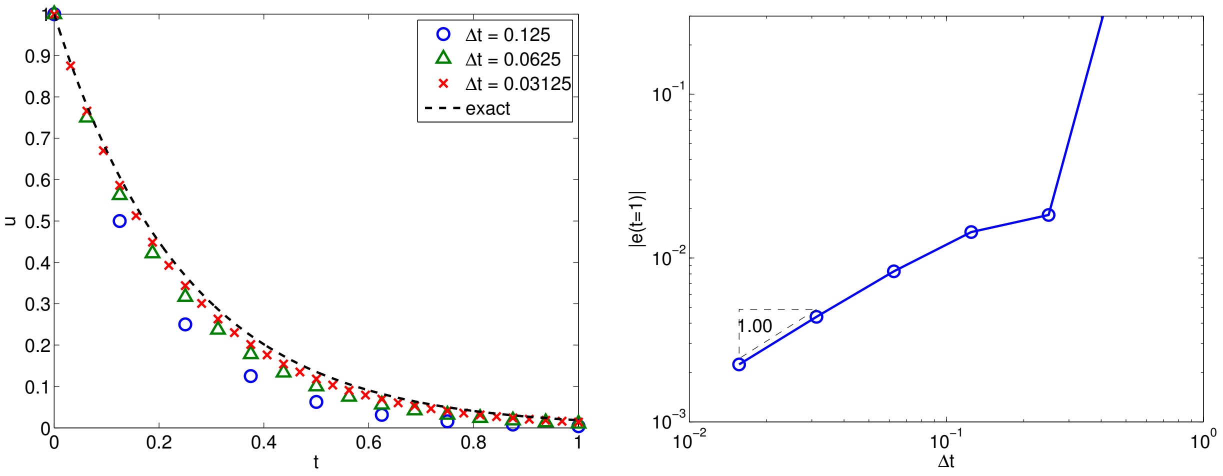
$$\Delta t \leq -\frac{2}{\lambda} \equiv \Delta t_{\text{cr}} \ .$$

(Note that since λ is negative we must reverse the inequality in $-2 \leq \lambda \Delta t$ upon division by λ.) Thus the Euler Forward scheme is absolutely stable only for $\Delta t \leq \Delta t_{\text{cr}}$: the scheme is *conditionally* absolutely stable.

We discuss the convergence behavior of the Euler Forward scheme for our model problem for the particular case in which $u_0 = 1$, $f(t) = 0$, and $\lambda = -4$ for $t_f = 1$. We can readily calculate that, for these parameter values, our critical time step for stability is $\Delta t_{\text{cr}} = -2/\lambda = 1/2$. For $\Delta t \leq \Delta t_{\text{cr}} = 1/2$, and in particular as $\Delta t \to 0$, we expect convergence. We furthermore anticipate, from our bound for the truncation error, (30), a convergence rate (order) of $p = 1$. We provide numerical evidence for these claims in Figure 4. In contrast, for $\Delta t > \Delta t_{\text{cr}}$, we expect instability. To demonstrate this claim, we consider $\Delta t = 1 \ (> \Delta t_{\text{cr}} = 1/2)$. In this case, the Euler Forward scheme (28)-(29), for our particular choice of parameters, reduces to

$$\tilde{u}^j = -3 \tilde{u}^{j-1} \ , \quad j = 1, \ldots, J \ , \tag{33}$$

$$\tilde{u}^0 = 1 \ . \tag{34}$$



(a) solution

(b) error

Figure 4: The convergence behavior for the Euler Forward approximation of our model problem for $u_0 = 1$, $f(t) = 0$, and $\lambda = -4$ for $t_f = 1$. Note $|e(t = 1)| \equiv |u(t^j) - \tilde{u}^j|$ at $t^j = j\Delta t = 1$.

The solution to this simple difference equation is $\tilde{u}^j = (-3)^j$. The numerical approximation grows exponentially in time — in contrast to the exact solution, which decays exponentially in time — and furthermore oscillates in sign — in contrast to the exact solution, which remains strictly positive. In short, the Euler Forward approximation is useless for $\Delta t > \Delta t_{\rm cr}$: Euler Forward estimates the solution at time level j from the derivative evaluated at time level $j-1$; for Δt too large, the extrapolation overshoots such that we realize growth rather than decay.

We should emphasize that the instability of the Euler Forward scheme for $\Delta t > \Delta t_{\rm cr}$ is *not* due to round-off errors (which involve "truncation," but not truncation of the variety discussed in this nutshell). In particular, all of our arguments related to stability — and the effects of instability through the amplification of truncation error — still obtain even in *infinite-precision arithmetic*. Indeed, there are no round-off errors or other finite-precision effects in our "by-hand" example of (33)-(34). Of course, an unstable difference equation will *also* amplify round-off errors, but that is an additional consideration and not the primary reason for the explosion.

5 Stiff Equations: Implicit *vs*. Explicit

We can understand the relative benefits of implicit and explicit schemes in the context of *stiff* equations — equations which exhibit disparate time scales. As our very simple example we shall consider our model problem with sinusoidal forcing,

$$\frac{du}{dt} = \lambda t + \cos(\omega t)\,, \quad 0 < t \le t_f\,, \tag{35}$$

$$u(0) = 0\,, \tag{36}$$

for the particular case in which $|\lambda| \gg \omega$ (and t_f is on the order of $1/\omega$, such that we consider at least one period of the sinusoidal source). The analytical solution $u(t)$ is given by (5) and presented in Figure 2. The short-time (transient, or homogeneous) response of the solution is dictated by the time constant $1/|\lambda|$; the long-time (steady-periodic, or inhomogenous) response is governed by the time constant $1/\omega \gg 1/|\lambda|$. We shall consider the case in which $\lambda = -100$ and $\omega = 4$; the transient and steady-periodic time scales differ by a factor of 25. We present in Figure 5 the Euler Backward and Euler Forward approximations, respectively, for three different values of the time step, Δt: $\Delta t = 0.015625$, $\Delta t = 0.0625$, and $\Delta t = 0.25$.

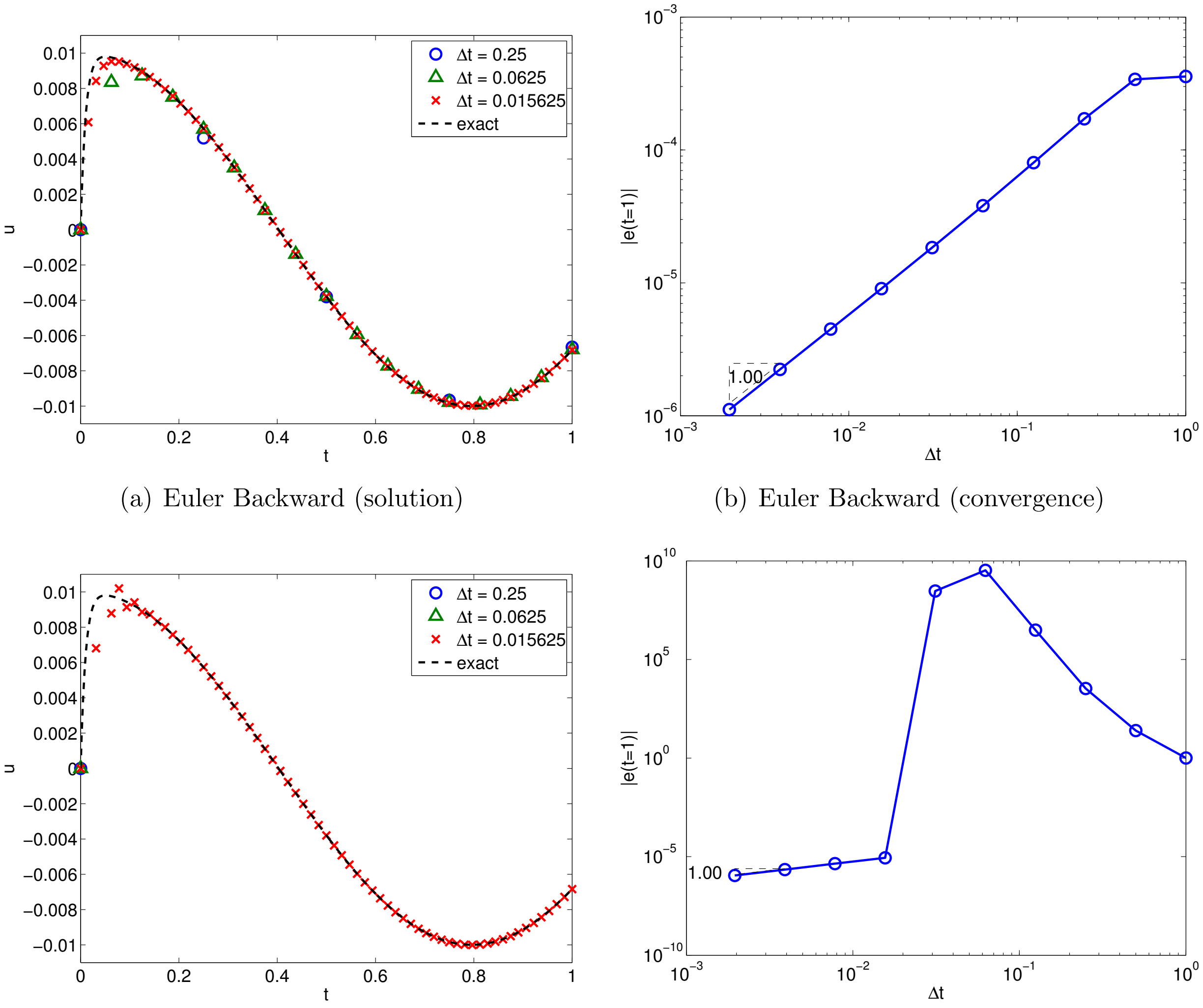
We first consider the results for the Euler Backward scheme. We recall that the Euler Backward scheme is stable for any time step Δt: the numerical results of Figure 5(a) confim that the Euler Backward approximation is, indeed, bounded for all time steps considered. For a large time step, in particular $\Delta t > 1/|\lambda|$, our Euler Backward approximation does not capture the initial transient, however it does still well represent the long-term sinusoidal behavior: Figure 5(b) confirms good accuracy. Thus, if the initial transient is not of interest, rather than choose a Δt associated with the characteristic time scale $1/|\lambda|$ — say $\Delta t = 0.01$ — we may select a Δt associated with the characteristic time scale $1/\omega$ — say $\Delta t = 0.1$. We can thereby significantly reduce the number of time steps, $t_f/\Delta t$.

We next turn to Euler Forward. We recall that the Euler Forward scheme is only conditionally stable: for our model problem, and for the particular parameter values associated with Figure 5, the critical time step is $\Delta t_{\text{cr}} = 2/|\lambda| = 0.02$. We may obtain, and we show in Figure 5(c), the Euler Forward numerical solution for only one of the three time steps proposed, $\Delta t = 1/64 < \Delta t_{\text{cr}}$, as the other two time steps considered ($\Delta t = 1/16$, $\Delta t = 1/4$) are greater than Δt_{cr}. Thus, even if we are not interested in the initial transient, we cannot select a large time step because the Euler Forward approximation will grow exponentially in time — clearly not relevant to the true sinusoidal behavior of the exact solution. The exponential growth of the error for $\Delta t > \Delta t_{\text{cr}}$ is clearly reflected in Figure 5(d).

CYAWTP 3. Consider (35)-(36) for the parameter values $\lambda = 100$, $\omega = 4$ and $t_f = 1$ considered in this section. For the case of Euler Backward discretization, how many time steps, J, are required to achieve an accuracy of $|e(t = 1)| = 0.0001$ (note that the amplitude of the solution is 0.01, and hence $|e(t = 1)| = 0.0001$ corresponds to a 1% error)? You may refer to Figure (5)(b). For the case of Euler Forward discretization, how many time steps, J, are required to achieve an accuracy of $|e(t = 1)| = 0.0001$?

CYAWTP 4. Consider (35)-(36) for the parameter values $\lambda = 100$, $\omega = 4$ and $t_f = 1$ considered in this section. Now assume that we are interested in both the transient behavior and the steady-periodic behavior. For the case of Euler Backward discretization with a *variable time step* — hence time levels $t^j, 0 \leq j \leq J$, which are not equispaced — how many time steps, J, are required to achieve an accuracy, for all t^j, $0 \leq j \leq J$, of roughly 0.0005? You may refer to Figure (5)(a) and assume (roughly true in this instance, though not always) that you may "jump" from the solution for one Δt to the solution for another Δt at any time at which the two errors are commensurate (without incurring any "residual" or additional error).

Stiff equations are ubiquitous in science and engineering. It is not uncommon to encounter problems for which the time scales may range over ten orders of magnitude. For example,



(a) Euler Backward (solution)

(b) Euler Backward (convergence)

(c) Euler Forward (solution)

(d) Euler Forward (convergence)

Figure 5: Application of the Euler Backward and Euler Forward schemes to a stiff equation. Note $|e(t=1)| \equiv |u(t^j) - \tilde{u}^j|$ at $t^j = j\Delta t = 1$.

the time scales associated with the dynamics of a passenger jet are many orders of magnitude larger than the time scales associated with the smallest eddies in the turbulent air flow outside the fuselage. In general, if the dynamics of the smallest time scale is not of interest, then an unconditionally stable scheme may be computationally advantageous: we may select the time step necessary to achieve sufficient accuracy without reference to any stability restriction. There are no explicit schemes which are unconditionally stable, and for this reason implicit schemes are often preferred for stiff equations.

We might conclude that explicit schemes serve very little purpose. In fact, this is not the case. In particular, we note that for Euler Backward, at every time step, we must effect a division operation, $1/(1-(\lambda\Delta t))$, whereas for Euler Forward we must effect a multiplication operation, $1+(\lambda\Delta t)$. Shortly we shall consider the extension of our methods to systems, often large systems, of many ODEs, in which the scalar algebraic operations of division and multiplication translate into matrix division and matrix multiplication, respectively. In general, matrix division — solution of linear systems of equations — is much more costly than matrix multiplication. Hence the total cost equation is more nuanced, as we now describe.

The computational effort can be expressed as the product of J, the number of time steps to reach the desired final time, t_f, and the "work" (or FLOPs) per time step. An implicit scheme will typically accommodate a large time step Δt, hence $J = t_f/\Delta t$ small, but demand considerable work per time step. In contrast, an explicit scheme will typically require a small time step Δt, hence $J = t_f/\Delta t$ large, but demand relatively little work per time step. For stiff equations in which the Δt informed by accuracy considerations is much, much larger than the Δt_{cr} dictated by stability considerations (for explicit schemes), implicit typically wins: fewer time steps, J, trumps more work per time step. On the other hand, for non-stiff equations, in which the Δt informed by accuracy considerations is of the same order as Δt_{cr} dictated by stability considerations (for explicit schemes), explicit often wins: since we shall select in any event a time step Δt which satisfies $\Delta t \leq \Delta t_{\mathrm{cr}}$, we might as well choose an explicit scheme to minimize the work per time step. Short summary: stiff implicit; non-stiff explicit.

6 Extensions

6.1 A Higher-Order Scheme

We have provided an example of an implicit scheme, Euler Backward, and an explicit scheme, Euler Forward. However, both these schemes are first-order. We provide here an example of a second-order scheme: the Crank-Nicolson method.

We again consider our model problem (3)-(4). The Crank-Nicolson approximation, $\tilde{u}^j \approx u(t^j)$, $0 \leq j \leq J$, satisfies

$$\frac{\tilde{u}^j - \tilde{u}^{j-1}}{\Delta t} = \frac{1}{2}(\,\lambda\tilde{u}^j + f(t^j)\,) + \frac{1}{2}(\,\lambda\tilde{u}^{j-1} + f(t^{j-1})\,)\,,\ j = 1,\ldots,J\,, \tag{37}$$

$$\tilde{u}^0 = u_0\,. \tag{38}$$

The Crank-Nicolson scheme is implicit. It can be shown that the scheme is unconditionally stable (for our model problem) and second-order accurate.

CYAWTP 5. Derive the Crank-Nicolson scheme by the integration approach: follow the same procedure as for Euler Backward and Euler Forward, but rather than the rectangle integration rule (right, and left, respectively), consider now for Crank-Nicolson the trapezoidal integration rule.

CYAWTP 6. Demonstrate that, for $\lambda < 0$, the Crank-Nicolson scheme is unconditionally stable: consider the homogeneous version of (37); demonstrate that the amplification factor, $\gamma \equiv |\tilde{u}^j|/|\tilde{u}^{j-1}|$, is less than or equal to unity for all positive Δt.

CYAWTP 7. Consider the Crank-Nicolson scheme for our model problem for $u_0 = 1$, $f(t) = 0$, and $\lambda = -2$ for $t_f = 1$. Find $\tilde{u}^J$ for $J = 1$, $J = 2$, $J = 4$, $J = 8$, and $J = 16$. How does $\tilde{u}^J$ compare to the exact solution, $\exp(-2)$, as you increase J (and hence decrease Δt)? What convergence rate $\tilde{u}^J \to \exp(-2)$ might you expect based on your knowledge of the trapezoidal rule of integration?

In general, high-order schemes — both implicit and explicit — can be quite efficient: a relatively modest increase in effort per time step (relative to low-order schemes), but considerably higher accuracy, at least for smooth problems. The Crank-Nicolson scheme is but one example of a high-order method: other examples, very popular, include the (implicit and explicit) Runge-Kutta multistage schemes.

6.2 General First-Order Scalar ODEs

At the outset we posed the general problem (1)-(2). We now return to this problem, and consider the application of the Euler Backward, Euler Forward, and Crank-Nicolson schemes to this much broader class of ODEs. The form of each scheme follows rather directly from the integration rule perspective. To wit, for Euler Backward (rectangle, right rule) we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t g(t^j, \tilde{w}^j)\,, \quad j = 1, \ldots, J\;; \tag{39}$$

for Euler Forward (rectangle, left rule) we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t g(t^{j-1}, \tilde{w}^{j-1})\,, \quad j = 1, \ldots, J\;; \tag{40}$$

and for Crank-Nicolson we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \frac{\Delta t}{2} g(t^j, \tilde{w}^j) + \frac{\Delta t}{2} g(t^{j-1}, \tilde{w}^{j-1})\,, \quad j = 1, \ldots, J\;. \tag{41}$$

Note in each case we must supplement the difference equation with our initial condition, $\tilde{w}^0 = w_0$.

6.3 Systems of First-Order ODEs

6.3.1 Formulation

We now consider a general system of n ODEs given by

$$\begin{aligned} \frac{dw}{dt} &= g(t, w)\,, \quad 0 \le t \le t_f\;, \\ w(0) &= w_0\;. \end{aligned} \tag{42}$$

Here w is an $n \times 1$ (column) vector of unknowns, $w = (w_1 \quad w_2 \quad \cdots \quad w_n)^{\mathrm{T}}$ — w is often referred to as the *state vector*, and the w_i, $1 \leq i \leq n$, as the state variables; $g(t,w)$ is an $n \times 1$ column vector of functions,

$$g(t,w) = (g_1(t,w) \quad g_2(t,w) \quad \cdots \quad g_n(t,w))^{\mathrm{T}} \, , \tag{43}$$

which represents the "dynamics"; and w_0 is an $n \times 1$ vector of initial conditions. In (42) we may admit any function $g(t,w)$ of nominal regularity.

Also of interest is the case of a linear system of first-order ODEs. In this case our $n \times 1$ vector $g(t,w)$ takes the form

$$g(t,w) = A(t)w + F(t) \, , \tag{44}$$

where $A(t)$ is an $n \times n$ matrix (hence $A(t)w$ is an $n \times 1$ vector, as required) and $F(t)$ is an $n \times 1$ vector. In this case, (42) reduces to

$$\frac{dw}{dt} = A(t)w + F(t) \ , \tag{45}$$

$$w(0) = w_0 \ .$$

The formulation (and analysis) further simplifies if $A(t)$ is time-independent, $A(t) = A$.

We now consider the application of the Euler Backward, Euler Forward, and Crank-Nicolson schemes to this system of equations. In fact, expressed in vector form, there is no change from the scalar case already presented in Section 6.2. Rather than repeat these equations verbatim, we instead present the system equations for the case of a *linear* system of ODEs with time-independent A: for Euler Backward we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t \, (\, A\tilde{w}^j + F(t^j) \,) \, , \quad j = 1, \ldots, J \ ; \tag{46}$$

for Euler Forward we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t \, (\, A\tilde{w}^{j-1} + F(t^{j-1}) \,) \, , \quad j = 1, \ldots, J \ ; \tag{47}$$

and for Crank-Nicolson we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \frac{\Delta t}{2} \, (\, A\tilde{w}^j + F(t^j) \,) + \frac{\Delta t}{2} \, (\, A\tilde{w}^{j-1} + F(t^{j-1}) \,) \, , \quad j = 1, \ldots, J \ . \tag{48}$$

Note in all cases we must supplement the difference equation with our initial condition, $\tilde{w}^0 = w_0$.

As always, Euler Backward and Crank-Nicolson are implicit schemes, whereas Euler Forward is an explicit scheme. We can now finally more explicitly identify the key computational difference between implicit and explicit schemes. For Euler Backward, (46), at each time level, to obtain $\tilde{w}^j$, we must solve a system of n linear equations in n unknowns :

$$(\, -\Delta t A + I \,)\tilde{w}^j = \tilde{w}^{j-1} + \Delta t \, F(t^j) \ ,$$

where I is the $n \times n$ identity matrix. In constrast, for Euler Forward, (47), at each time level, to obtain $\tilde{w}^j$, we need only evaluate the product of an $n \times n$ matrix and an $n \times 1$ vector.

Hence our earlier claim that, in general, an implicit method will require more work per time step than an explicit method.

We now summarize the performance of these schemes. The Euler Backward scheme (46) is unconditionally absolutely stable for ODEs which are stable (in the sense that the eigenvalues of A are all of non-positive real part). However, the Euler Backward scheme is only first-order accurate, and hence often not the best choice. The Euler Forward scheme, (47), is only conditionally absolutely stable, and furthermore only first-order accurate. The Euler Forward scheme is almost never the best choice: we include it here in this nutshell primarily for illustrative value, but also because it often appears as a "building block" in more advanced numerical approaches. Much better explicit schemes, appropriate for non-stiff equations, are available — larger Δt_{cr} and higher-order accuracy: Runge-Kutta schemes are the most popular. Finally, the Crank-Nicolson scheme, (48), is unconditionally absolute stable for ODEs which are stable. (Note, however, that Crank-Nicolson can exhibit amplification factors which approach unity, which can on occasion create difficulties.) The Crank-Nicolson scheme is also second-order accurate, and hence often a very good choice in practice. Good alternatives, with somewhat better stability characteristics and also higher-order accuracy, include implicit Runge-Kutta methods.

6.3.2 Reduction to First-Order Form

We begin our discussion with the classical second-order harmonic oscillator so ubiquitous in engineering applications:

$$m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky = f(t), \quad 0 < t < t_f \ ,$$

$$y(0) = y_0\,, \quad \frac{dy}{dt}(0) = \dot{y}_0 \ .$$

This second-order ODE governs the oscillations of (say) a spring-mass-damper system as a function of time t: y is the displacement, m is the lumped mass, c is the damping constant, k is the spring constant, and f represents the external forcing. From a mathematical perspective, this IVP ODE is *second-order* — we thus require *two* initial conditions, one for the displacement, and one for the velocity — and also *linear*.

It is possible to directly numerically tackle this second-order system, for example with Newmark integration schemes. However, we shall prefer here to reduce our second-order IVP ODE to a system of two first-order IVP ODEs such that we can then directly apply the general technology developed in the previous sections — and by the computational community — for systems of first-order IVP ODEs. The transformation from second-order IVP ODE to two first-order IVP ODEs is very simple.

We first choose our state variables as

$$w_1(t) = y(t) \quad \text{and} \quad w_2(t) = \frac{dy}{dt}(t) \ ,$$

corresponding to the displacement and velocity, respectively. We directly obtain the trivial relationship between w_1 and w_2

$$\frac{dw_1}{dt} = \frac{dy}{dt} = w_2 \ .$$

Furthermore, the governing second-order ODE can be rewritten in terms of w_1 and w_2 as

$$\frac{dw_2}{dt} = \frac{d}{dt}\frac{dy}{dt} = \frac{d^2y}{dt^2} = -\frac{b}{m}\frac{dy}{dt} - \frac{k}{m}y + \frac{1}{m}f(t) = -\frac{b}{m}w_2 - \frac{k}{m}w_1 + \frac{1}{m}\,f(t) \ .$$

We can thus rewrite the original second-order ODE as a system of two first-order ODEs,

$$\frac{d}{dt}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} w_2 \\ -\frac{k}{m}w_1 - \frac{b}{m}w_2 + \frac{1}{m}\,f \end{pmatrix} .$$

This equation can be written in the matrix form as

$$\frac{d}{dt}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{pmatrix}}_{A}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ \frac{1}{m}\,f \end{pmatrix}}_{F} \tag{49}$$

with the initial condition $w_1(0) = y_0, w_2(0) = \dot{y}_0$. If we now define $w = (w_1 \quad w_2)^{\mathrm{T}}$ we recognize that (49) is precisely of the desired first-order system form, (45).

CYAWTP 8. Consider the nonlinear second-order IVP ODE given by $\ddot{y} + y^3 = f(t)$, $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$. Reduce this equation to a system of two first-order IVP ODEs of the form (42): identify the state vector; the "dynamics" function (vector) g; and the initial conditions.

CYAWTP 9. Consider the third-order linear IVP ODE given by $-\dddot{z} + \ddot{z} - \dot{z} = f(t)$, $z(0) = z_0$, $\dot{z}(0) = \dot{z}_0$, $\ddot{z}(0) = \ddot{z}_0$. Reduce this equation to a linear system of three first-order IVP ODEs of the form (45): identify the state vector; the elements of A and F; and the initial condition vector.

To close we consider a more general case: $n/2$ coupled oscillators (for n an even integer); the oscillators may represent different degrees of freedom in a large system. These coupled oscillators can be described by the set of equations

$$\begin{aligned}
\frac{d^2y^{(1)}}{dt^2} &= \mathcal{F}^{(1)}\left(\frac{dy^{(j)}}{dt}, y^{(j)},\ 1 \le j \le n/2\right) + f^{(1)}(t) \ , \\
\frac{d^2y^{(2)}}{dt^2} &= \mathcal{F}^{(2)}\left(\frac{dy^{(j)}}{dt}, y^{(j)},\ 1 \le j \le n/2\right) + f^{(2)}(t) \ , \\
&\ \ \vdots \\
\frac{d^2y^{(n/2)}}{dt^2} &= \mathcal{F}^{(n/2)}\left(\frac{dy^{(j)}}{dt}, y^{(j)},\ 1 \le j \le n/2\right) + f^{(n/2)}(t) \ ,
\end{aligned} \tag{50}$$

where $\mathcal{F}^{(j)}, 1 \leq j \leq n/2$, are specified functions.

We first convert this system of equations to state space form. We identify

$$\begin{aligned} w_1 &= y^{(1)}, & w_2 &= \frac{dy^{(1)}}{dt}\ , \\ w_3 &= y^{(2)}, & w_4 &= \frac{dy^{(2)}}{dt}\ , \\ &\ \ \vdots \\ w_{n-1} &= y^{(n/2)}, & w_n &= \frac{dy^{(n/2)}}{dt}\ , \end{aligned}$$

from which we form our $n \times 1$ state vector. We can then express (50) as (42), where the $n \times 1$ vector $g(t, w)$ is given by

$$g_i(t,w) = \begin{cases} w_{i+1} & i = 1, 3, \dots, n-1 \\ \mathcal{F}^{(i/2)} \text{ (expressed in terms of } w) & i = 2, 4, \dots, n \end{cases}\ ,$$

and w_0 is the $n \times 1$ initial condition vector expressed in terms of the (two) initial conditions on each of the $y^{(j)}$, $1 \leq j \leq n/2$,

$$w_0 = \left(y^{(1)}(0) \quad \frac{dy^{(1)}}{dt}(0) \quad y^{(2)}(0) \quad \frac{dy^{(2)}}{dt}(0) \quad \dots \quad y^{(n/2)}(0) \quad \frac{dy^{(n/2)}}{dt}(0) \right)^{\mathrm{T}} .$$

This same procedure readily specializes to linear systems in the case in which $g(t, w)$ takes the form (44).

CYAWTP 10. Consider two beads each of mass m on a string under tension T. The string is fixed at $x = 0$ and $x = L$ and the two beads are positioned horizontally at $x = L/3$ and $x = 2L/3$, respectively. We denote respectively by $y^{(1)}(t)$ and $y^{(2)}(t)$ the vertical position of the first bead and the second bead as a function of time t. Under the assumption of small-angle displacements (and negligible damping) the system is described by the pair ($n/2 = 2$) of coupled oscillators

$$m\frac{d^2y^{(1)}}{dt^2} + \frac{3T}{L}(2y^{(1)} - y^{(2)}) = f^{(1)}(t)$$

$$m\frac{d^2y^{(2)}}{dt^2} + \frac{3T}{L}(2y^{(2)} - y^{(1)}) = f^{(2)}(t)\ ,$$

with initial conditions $y^{(1)}(0) = y_0^{(1)}$, $\dot{y}^{(1)}(0) = \dot{y}_0^{(1)}$, $y^{(2)}(0) = y_0^{(2)}$, $\dot{y}^{(2)}(0) = \dot{y}_0^{(2)}$. Here $f^{(1)}(t)$ and $f^{(2)}(t)$ denote the applied vertical force on the first bead and the second bead, respectively. Reduce this equation to a linear system of four first-order IVP ODEs of the form (45): identify the state vector; the elements of A and F; and the initial condition vector.

7 Perspectives

We have provided here only a first look at the topic of integration of IVP ODEs. A more in-depth study may be found in *Math, Numerics, and Programming (for Mechanical Engineers)*, M Yano, JD Penn, G Konidaris, and AT Patera, available on MIT OpenCourseWare; this text adopts similar notation to these nutshells, and hence can serve as a companion reference. For an even more comprehensive view, from both the computational and theoretical perspectives, we recommend *Numerical Mathematics*, A Quarteroni, R Sacco, F Saleri, Springer, 2000.

We briefly comment on two related topics.

1. We have focused here exclusively on IVP ODEs: *initial value problems.* In fact, BVP ODEs also appear frequently in engineering analysis: *boundary value problems.* Although BVP ODEs do share some features, in particular related to finite difference discretization, with IVP ODEs, BVPs are fundamentally different from IVPs as regards both theoretical foundations and also computational procedures.

2. We have discussed here exclusively ODEs: *ordinary* differential equations. *Partial* differential equations (PDEs) also appear frequently in engineering analysis. In fact, the methods we present in this nutshell are rather directly relevant to the approximation of PDEs, in particular evolution equations: we may think of A in (45) as a discretization of the "spatial" part of the partial differential operator.

   We note that, for ODEs, essentially all consistent schemes — even schemes which are conditionally stable — are convergent: as Δt tends to zero we will perforce satisfy $\Delta t \leq \Delta t_{\mathrm{cr}}$. However, for PDEs, the situation is different: the critical time step may depend on the spatial discretization (say Δx, inversely proportional to n), and if Δt and Δx do not tend to zero in the proper ratio then convergence will not be realized. On a related note, all partial differential equations are, in effect, stiff: this places a premium on implicit approaches, and on efficient solver strategies which exploit, for example, the sparsity and structure of A.

Both of these topics are discussed in considerable depth in *Numerical Mathematics*, A Quarteroni, R Sacco, F Saleri, Springer, 2000.

8 Appendices

8.1 Euler Backward: Consistency

Let us now analyze the consistency of the Euler Backward integration scheme. We shall assume that our solution u of (3) is twice continuously differentiable with respect to time. To start, we construct a Taylor expansion for $u(t^{j-1})$ about t^j,

$$u(t^{j-1}) = u(t^j) - \Delta t \frac{du}{dt}(t^j) - \underbrace{\int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \frac{d^2u}{dt^2}(s)\, ds \right) d\tau}_{s^j(u)} \ . \tag{51}$$

The expansion (51) is simple to derive. First, by the fundamental theorem of calculus,

$$\int_{t^{j-1}}^{\tau} \frac{d^2u}{dt^2}(s)\,ds = \frac{du}{dt}(\tau) - \frac{du}{dt}(t^{j-1}) \ . \tag{52}$$

Integrating both sides of (52) over the interval (t^{j-1}, t^j),

$$\begin{aligned}
\int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \frac{d^2u}{dt^2}(s)\,ds \right) d\tau &= \int_{t^{j-1}}^{t^j} \left(\frac{du}{dt}(\tau) \right) d\tau - \int_{t^{j-1}}^{t^j} \left(\frac{du}{dt}(t^{j-1}) \right) d\tau \\
&= u(t^j) - u(t^{j-1}) - (t^j - t^{j-1}) \frac{du}{dt}(t^{j-1}) \\
&= u(t^j) - u(t^{j-1}) - \Delta t \frac{du}{dt}(t^{j-1}) \ . \qquad (53)
\end{aligned}$$

If we now in (53) move $u(t^{j-1})$ to the left-hand side and $s^j(u)$ to the right-hand side we directly obtain (51).

We now subsitute our Taylor expansion (51) into the expression for the truncation error,

$$\tau^j_{\text{trunc}} \equiv \frac{u(t^j) - u(t^{j-1})}{\Delta t} - \lambda u(t^j) - f(t^j) \ , \ j = 1, \ldots, J \ ,$$

to obtain

$$\begin{aligned}
\tau^j_{\text{trunc}} &= \frac{1}{\Delta t} \left(u(t^j) - \left(u(t^j) - \Delta t \frac{du}{dt}(t^j) - s^j(u) \right) \right) - \lambda u(t^j) - f(t^j) \\
&= \underbrace{\frac{du}{dt}(t^j) - \lambda u(t^j) - f(t^j)}_{=0 \text{ from our ODE, (3)}} + \frac{s^j(u)}{\Delta t} \\
&= \frac{s^j(u)}{\Delta t} \quad j = 1, \ldots, J \ .
\end{aligned}$$

We now bound the remainder term $s^j(u)$ as a function of Δt. In particular, we note that

$$\begin{aligned}
s^j(u) = \int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \frac{d^2u}{dt^2}(s)\,ds \right) d\tau &\le \int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \left| \frac{d^2u}{dt^2}(s) \right| ds \right) d\tau \\
&\le \max_{t \in [t^{j-1}, t^j]} \left| \frac{d^2u}{dt^2}(t) \right| \int_{t^{j-1}}^{t^j} \int_{t^{j-1}}^{\tau} ds\, d\tau \\
&= \max_{t \in [t^{j-1}, t^j]} \left| \frac{d^2u}{dt^2}(t) \right| \frac{\Delta t^2}{2}, \quad j = 1, \ldots, J \ .
\end{aligned}$$

Hence we may bound the maximum truncation error as

$$\tau_{\text{trunc}}^{\max} = \max_{j=1,\ldots,J} |\tau_{\text{trunc}}^j| \le \max_{j=1,\ldots,J} \left(\frac{1}{\Delta t} \max_{t \in [t^{j-1}, t^j]} \left| \frac{d^2u}{dt^2}(t) \right| \frac{\Delta t^2}{2} \right) \le \frac{\Delta t}{2} \max_{t \in [0, t_f]} \left| \frac{d^2u}{dt^2}(t) \right| \ .$$

It directly follows that

$$\tau_{\text{trunc}}^{\max} \le \frac{\Delta t}{2} \max_{t \in [0, t_f]} \left| \frac{d^2u}{dt^2}(t) \right| \to 0 \ \text{ as } \ \Delta t \to 0 \ , \tag{54}$$

and hence the Euler Backward scheme (8) is consistent with our ODE (3).

8.2 Euler Backward: Convergence

Let us denote the solution error by e^j,

$$e^j \equiv u(t^j) - \tilde{u}(t^j) \ .$$

We first relate the evolution of the error to the truncation error. To begin, we recall that

$$u(t^j) - u(t^{j-1}) - \lambda \Delta t \, u(t^j) - \Delta t \, f(t^j) = \Delta t \, \tau_{\text{trunc}}^j \ ,$$

$$\tilde{u}^j - \tilde{u}^{j-1} - \lambda \Delta t \, \tilde{u}^j - \Delta t f(t^j) = 0 \ ;$$

subtracting these two equations and recalling the definition of the error, $e^j \equiv u(t^j) - \tilde{u}^j$, we obtain

$$e^j - e^{j-1} - \lambda \Delta t \, e^j = \Delta t \, \tau_{\text{trunc}}^j \ ,$$

or, upon rearrangment,

$$(1 - \lambda \Delta t) e^j - e^{j-1} = \Delta t \, \tau_{\text{trunc}}^j \ . \tag{55}$$

We see that the solution error itself satisfies an Euler Backward difference equation, but now with the truncation error as the source term. Furthermore, since $u(0) = \tilde{u}^0 = u_0$, $e^0 = 0$. It follows that, if the truncation error τ_{trunc}^j is zero at all time levels, then the solution error is also zero at all time levels.

However, in general, the truncation error is nonzero. To proceed, we multiply (55) by $(1 - \lambda \Delta t)^{j-1}$ to obtain

$$(1 - \lambda \Delta t)^j e^j - (1 - \lambda \Delta t)^{j-1} e^{j-1} = (1 - \lambda \Delta t)^{j-1} \Delta t \, \tau_{\text{trunc}}^j \ . \tag{56}$$

We now sum (56) for $j = 1, \ldots, k$, for some $k \le J$,

$$\sum_{j=1}^{k} \left[(1 - \lambda \Delta t)^j e^j - (1 - \lambda \Delta t)^{j-1} e^{j-1} \right] = \sum_{j=1}^{k} \left[(1 - \lambda \Delta t)^{j-1} \Delta t \, \tau_{\text{trunc}}^j \right] .$$

This is a telescopic series and all the middle terms on the left-hand side cancel. More explicitly, the sum of the k equations

$$
\begin{aligned}
(1-\lambda\Delta t)^k e^k - (1-\lambda\Delta t)^{k-1} e^{k-1} &= (1-\lambda\Delta t)^{k-1}\Delta t\,\tau^k_{\text{trunc}}\\
(1-\lambda\Delta t)^{k-1} e^{k-1} - (1-\lambda\Delta t)^{k-2} e^{k-2} &= (1-\lambda\Delta t)^{k-2}\Delta t\,\tau^{k-1}_{\text{trunc}}\\
&\vdots\\
(1-\lambda\Delta t)^2 e^2 - (1-\lambda\Delta t)^1 e^1 &= (1-\lambda\Delta t)^1\Delta t\,\tau^2_{\text{trunc}}\\
(1-\lambda\Delta t)^1 e^1 - (1-\lambda\Delta t)^0 e^0 &= (1-\lambda\Delta t)^0\Delta t\,\tau^1_{\text{trunc}}
\end{aligned}
$$

simplifies to

$$(1-\lambda\Delta t)^k e^k - e^0 = \sum_{j=1}^{k}(1-\lambda\Delta t)^{j-1}\Delta t\,\tau^j_{\text{trunc}}\ .$$

We now recall that $\tilde{u}^0 = u(0)$, and hence the initial error is zero ($e^0 = 0$). We are thus left with

$$(1-\lambda\Delta t)^k e^k = \sum_{j=1}^{k}(1-\lambda\Delta t)^{j-1}\Delta t\,\tau^j_{\text{trunc}}$$

or, equivalently,

$$e^k = \sum_{j=1}^{k}(1-\lambda\Delta t)^{j-k-1}\Delta t\,\tau^j_{\text{trunc}}\ . \tag{57}$$

It remains to bound this sum.

Towards that end, we recall that $\tau^{\max}_{\text{trunc}} \equiv \max_{j=1,\ldots,J}|\tau^j_{\text{trunc}}|$, and thus from (57) we obtain

$$|e^k| \le \Delta t\,\tau^{\max}_{\text{trunc}}\sum_{j=1}^{k}(1-\lambda\Delta t)^{j-k-1}\ . \tag{58}$$

We now recall the amplification factor for the Euler Backward scheme, $\gamma = 1/(1-\lambda\Delta t)$, in terms of which we may write

$$
\begin{aligned}
\sum_{j=1}^{k}(1-\lambda\Delta t)^{j-k-1} &= \frac{1}{(1-\lambda\Delta t)^k} + \frac{1}{(1-\lambda\Delta t)^{k-1}} + \cdots + \frac{1}{(1-\lambda\Delta t)}\\
&= \gamma^k + \gamma^{k-1} + \cdots + \gamma\ .
\end{aligned}
$$

Stability is now invoked: because the scheme is absolutely stable, the amplification factor satisfies $\gamma \leq 1$. It thus follows that

$$\sum_{j=1}^{k} (1 - \lambda \Delta t)^{j-k-1} = \gamma^k + \gamma^{k-1} + \cdots + \gamma \leq k\gamma \leq k \ .$$

We now insert this result into (58) to obtain

$$|e^k| \leq (k\Delta t)\, \tau_{\text{trunc}}^{\max} = t^k\, \tau_{\text{trunc}}^{\max} \ .$$

Finally, we invoke consistency: $\tau_{\text{trunc}}^{\max} \to 0$ as $\Delta t \to 0$. Thus,

$$|e^k| \leq t^k\, \tau_{\text{trunc}}^{\max} \quad \to 0 \quad \text{as} \quad \Delta t \to 0$$

for fixed $t^k = k\Delta t$. Note that the proof of convergence relies on stability ($\gamma \leq 1$) and consistency ($\tau_{\text{trunc}}^{\max} \to 0$ as $\Delta t \to 0$).

We may now invoke our specific expression for the truncation error, (54), to develop the bound

$$|e^k| \leq t^k \frac{\Delta t}{2} \max_{t \in [0, t_f]} \left| \frac{d^2 u}{dt^2}(t) \right| \ . \qquad (59)$$

for fixed $t^k = k\Delta t$. (We may also replace t^k by t_f to develop a uniform bound over the interval $[0, t_f]$.) We remark that the bound (59) can, in general, be pessimistic: since $\gamma < 1$ for Euler Backward, the error at time t is affected only by the truncation errors, and hence the second derivative of u, for times t' ($\leq t$) relatively close to t; the "proof" is (57).