DRAFT V1.2

From

Math, Numerics, & Programming

(for Mechanical Engineers)

Masayuki Yano
James Douglass Penn
George Konidaris
Anthony T Patera

September 2012

Contents

Unit IV

(Numerical) Differential Equations

Chapter 20

Motivation

Although mobile robots operating in flat, indoor environments can often perform quite well without any suspension, in uneven terrain, a well-designed suspension can be critical.

An actual robot suspension and its simplified model are shown in Figure 20.1. The rear and front springs with spring constants k_1 and k_2 serve to decouple the rest of the robot chassis from the wheels, allowing the chassis and any attached instrumentation to "float" relatively unperturbed while the wheels remain free to follow the terrain and maintain traction. The rear and front dampers with damping coefficients c_1 and c_2 (shown here inside the springs) dissipate energy to prevent excessive chassis displacements (e.g., from excitation of a resonant mode) and oscillations. Note that in our "half-robot" model, k_1 accounts for the *combined* stiffness of both rear wheels, and k_2 accounts for the combined stiffness of both front wheels. Similarly, c_1 and c_2 account for the combined damping coefficients of both rear wheels and both front wheels, respectively.

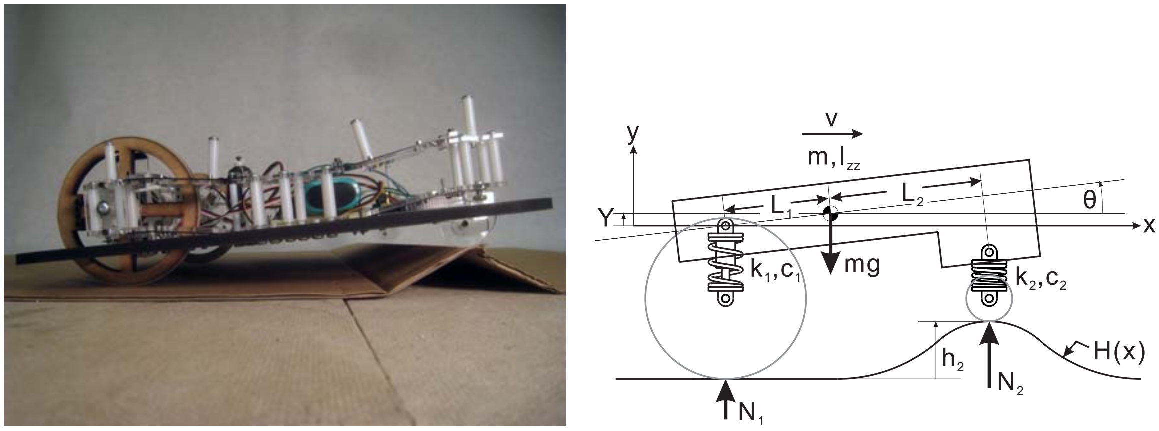
We are particularly concerned with the possibility of either the front or rear wheels losing contact with the ground, the consequences of which — loss of control and a potentially harsh landing — we wish to avoid.

To aid in our understanding of robot suspensions and, in particular, to understand the conditions resulting in loss of contact, we wish to develop a simulation based on the simple model of Figure 20.1(b). Specifically, we wish to simulate the transient (time) response of the robot with suspension traveling at some constant velocity v over a surface with profile $H(x)$, the height of the ground as a function of x, and to check if loss of contact occurs. To do so, we must integrate the differential equations of motion for the system.

First, we determine the motion at the rear (subscript 1) and front (subscript 2) wheels in order to calculate the normal forces N_1 and N_2. Because we assume constant velocity v, we can determine the position in x of the center of mass at any time t (we assume $X(t = 0) = 0$) as

$$X = vt \; . \tag{20.1}$$

Given the current state Y, $\dot{Y}$, θ (the inclination of the chassis), and $\dot{\theta}$, we can then calculate the



(a) Actual robot suspension.

(b) Robot suspension model.

Courtesy of James Penn. Used with permission.

Figure 20.1: Mobile robot suspension

positions and velocities in both x and y at the rear and front wheels (assuming θ is small) as

$$
\begin{aligned}
X_1 &= X - L_1, \quad (\dot{X}_1 = v) \ , \\
X_2 &= X + L_2, \quad (\dot{X}_2 = v) \ , \\
Y_1 &= Y - L_1\theta \ , \\
\dot{Y}_1 &= \dot{Y} - L_1\dot{\theta} \ , \\
Y_2 &= Y + L_2\theta \ , \\
\dot{Y}_2 &= \dot{Y} + L_2\dot{\theta} \ ,
\end{aligned}
\tag{20.2}
$$

where L_1 and L_2 are the distances to the system's center of mass from the rear and front wheels. (Recall $\dot{}$ refers to time derivative.) Note that we define $Y = 0$ as the height of the robot's center of mass with both wheels in contact with flat ground and both springs at their unstretched and uncompressed lengths, i.e., when $N_1 = N_2 = 0$. Next, we determine the heights of the ground at the rear and front contact points as

$$
\begin{aligned}
h_1 &= H(X_1) \ , \\
h_2 &= H(X_2) \ .
\end{aligned}
\tag{20.3}
$$

Similarly, the rates of change of the ground height at the rear and front are given by

$$
\begin{aligned}
\frac{dh_1}{dt} &= \dot{h}_1 = v\frac{d}{dx}H(X_1) \ , \\
\frac{dh_2}{dt} &= \dot{h}_2 = v\frac{d}{dx}H(X_2) \ .
\end{aligned}
\tag{20.4}
$$

Note that we must multiply the spatial derivatives $\frac{dH}{dx}$ by $v = \frac{dX}{dt}$ to find the temporal derivatives.

While the wheels are in contact with the ground we can determine the normal forces at the rear and front from the constitutive equations for the springs and dampers as

$$
\begin{aligned}
N_1 &= k_1(h_1 - Y_1) + c_1(\dot{h}_1 - \dot{Y}_1) \ , \\
N_2 &= k_2(h_2 - Y_2) + c_2(\dot{h}_2 - \dot{Y}_2) \ .
\end{aligned}
\tag{20.5}
$$

If either N_1 or N_2 is calculated from Equations (20.5) to be less than or equal to zero, we can determine that the respective wheel has lost contact with the ground and stop the simulation, concluding loss of contact, i.e., failure.

Finally, we can determine the rates of change of the state from the linearized ($\cos\theta \approx 1$, $\sin\theta \approx \theta$) equations of motion for the robot, given by Newton-Euler as

$$\begin{aligned} \ddot{X} &= 0, \quad \dot{X} = v, \quad X(0) = 0 \ , \\ \ddot{Y} &= -g + \frac{N_1 + N_2}{m}, \quad \dot{Y}(0) = \dot{Y}_0, \quad Y(0) = Y_0 \ , \\ \ddot{\theta} &= \frac{N_2 L_2 - N_1 L_1}{I_{\mathrm{zz}}}, \quad \dot{\theta}(0) = \dot{\theta}_0, \quad \theta(0) = \theta_0 \ , \end{aligned} \tag{20.6}$$

where m is the mass of the robot, and I_{zz} is the moment of inertia of the robot about an axis parallel to the Z axis passing through the robot's center of mass.

In this unit we shall discuss the numerical procedures by which to integrate systems of ordinary differential equations such as (20.6). This integration can then permit us to determine loss of contact and hence failure.

Chapter 21

Initial Value Problems

21.1 Scalar First-Order Linear ODEs

21.1.1 Model Problem

Let us consider a canonical initial value problem (IVP),

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

$$u(0) = u_0 \ .$$

The objective is to find u over all time $t \in \,]0, t_f]$ that satisfies the ordinary differential equation (ODE) and the initial condition. This problem belongs to the class of *initial value problems* (IVP) since we supplement the equation with condition(s) only at the initial time. The ODE is *first order* because the highest derivative that appears in the equation is the first-order derivative; because it is first order, only one initial condition is required to define a unique solution. The ODE is *linear* because the expression is linear with respect to u and its derivative du/dt; note that f does not have to be a linear function of t for the ODE to be linear. Finally, the equation is *scalar* since we have only a single unknown, $u(t) \in \mathbb{R}$. The coefficient $\lambda \in \mathbb{R}$ controls the behavior of the ODE; $\lambda < 0$ results in a stable (i.e. decaying) behavior, whereas $\lambda > 0$ results in an unstable (i.e. growing) behavior.

We can motivate this model problem (with $\lambda < 0$) 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 (away from the body) 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 body "diameter" in the numerator, 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 our ordinary differential equation (ODE) initial value problem (IVP), with $\lambda = -h \, \text{Area}/\rho c \, \text{Vol}$ and $f(t) = \dot{q}(t)/\rho c \, \text{Vol}$, where ρ and c are the body density and specific heat, respectively, and Area and Vol are the body surface area and volume, respectively.

In fact, it is possible to express the solution to our model problem in closed form (as a convolution). Our interest in the model problem is thus not because we require a numerical solution procedure for this particular simple problem. Rather, as we shall see, our model problem will provide a foundation on which to construct and understand numerical procedures for much more difficult problems — which do *not* admit closed-form solution.

21.1.2 Analytical Solution

Before we pursue numerical methods for solving the IVP, let us study the analytical solution for a few cases which provide insight into the solution and also suggest test cases for our numerical approaches.

Homogeneous Equation

The first case considered is the homogeneous case, i.e., $f(t) = 0$. Without loss of generality, let us set $u_0 = 1$. Thus, we consider

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

$$u(0) = 1 \ .$$

We find the analytical solution by following the standard procedure for obtaining the homogeneous solution, i.e., substitute $u = \alpha e^{\beta t}$ to obtain

$$(\mathrm{LHS}) = \frac{du}{dt} = \frac{d}{dt}(\alpha e^{\beta t}) = \alpha \beta e^{t} \ ,$$

$$(\mathrm{RHS}) = \lambda \alpha e^{\beta t} \ .$$

Equating the LHS and RHS, we obtain $\beta = \lambda$. The solution is of the form $u(t) = \alpha e^{\lambda t}$. The coefficient α is specified by the initial condition, i.e.

$$u(t = 0) = \alpha = 1 \ ;$$

thus, the coefficient is $\alpha = 1$. The solution to the homogeneous ODE is

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

Note that solution starts from 1 (per the initial condition) and decays to zero for $\lambda < 0$. The decay rate is controlled by the time constant $1/|\lambda|$ — the larger the λ, the faster the decay. The solution for a few different values of λ are shown in Figure 21.1.

We note that for $\lambda > 0$ the solution grows exponentially in time: the system is unstable. (In actual fact, in most physical situations, at some point additional terms — for example, nonlinear effects not included in our simple model — would become important and ensure saturation in some steady state.) In the remainder of this chapter *unless specifically indicated otherwise* we shall assume that $\lambda < 0$.

Constant Forcing

Next, we consider a constant forcing case with $u_0 = 0$ and $f(t) = 1$, i.e.

$$\frac{du}{dt} = \lambda u + 1 \ ,$$

$$u_0 = 0 \ .$$

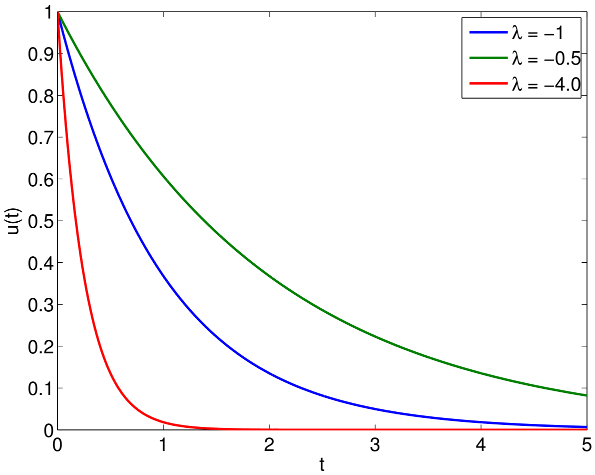


Figure 21.1: Solutions to the homogeneous equation.

We have already found the homogeneous solution to the ODE. We now find the particular solution. Because the forcing term is constant, we consider a particular solution of the form $u_p(t) = \gamma$. Substitution of u_p yields

$$0 = \lambda\gamma + 1 \quad \Rightarrow \quad \gamma = -\frac{1}{\lambda} \ .$$

Thus, our solution is of the form

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

Enforcing the initial condition,

$$u(t=0) = -\frac{1}{\lambda} + \alpha = 0 \quad \Rightarrow \quad \alpha = \frac{1}{\lambda} \ .$$

Thus, our solution is given by

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

The solutions for a few different values of λ are shown in Figure 21.2. For $\lambda < 0$, after the transient which decays on the time scale $1/|\lambda|$, the solution settles to the steady state value of $-1/\lambda$.

Sinusoidal Forcing

Let us consider a final case with $u_0 = 0$ and a sinusoidal forcing, $f(t) = \cos(\omega t)$, i.e.

$$\frac{du}{dt} = \lambda u + \cos(\omega t) \ ,$$

$$u_0 = 0 \ .$$

Because the forcing term is sinusoidal with the frequency ω, the particular solution is of the form $u_p(t) = \gamma\sin(\omega t) + \delta\cos(\omega t)$. Substitution of the particular solution to the ODE yields

$$(\text{LHS}) = \frac{du_p}{dt} = \omega(\gamma\cos(\omega t) - \delta\sin(\omega t)) \ ,$$

$$(\text{RHS}) = \lambda(\gamma\sin(\omega t) + \delta\cos(\omega t)) + \cos(\omega t) \ .$$

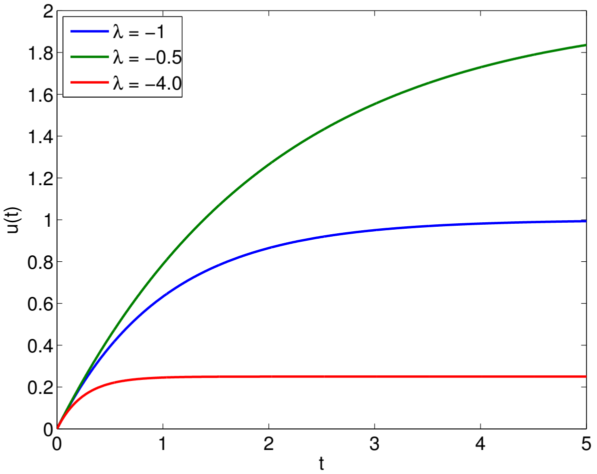


Figure 21.2: Solutions to the ODE with unit constant forcing.

Equating the LHS and RHS and collecting like coefficients we obtain

$$\omega\gamma = \lambda\delta + 1 \ ,$$

$$-\omega\delta = \lambda\gamma \ .$$

The solution to this linear system is given by $\gamma = \omega/(\omega^2+\lambda^2)$ and $\delta = -\lambda/(\omega^2+\lambda^2)$. Thus, the solution is of the form

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

Imposing the boundary condition, we obtain

$$u(t=0) = -\frac{\lambda}{\omega^2+\lambda^2} + \alpha = 0 \quad \Rightarrow \quad \alpha = \frac{\lambda}{\omega^2+\lambda^2} \ .$$

Thus, the solution to the IVP with the sinusoidal forcing is

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

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 21.3. The steady state behavior is controlled by the sinusoidal forcing function and has the time scale of $1/\omega$. On the other hand, the initial transient is controlled by λ and has the time scale of $1/|\lambda|$. In particular, note that for $|\lambda| \gg \omega$, the solution exhibits very different time scales in the transient and in the steady (periodic) state. This is an example of a *stiff equation* (we shall see another example at the conclusion of this unit). Solving a stiff equation introduces additional computational challenges for numerical schemes, as we will see shortly.

21.1.3 A First Numerical Method: Euler Backward (Implicit)

In this section, we consider the Euler Backward integrator for solving initial value problems. We first introduce the time stepping scheme and then discuss a number of properties that characterize the scheme.

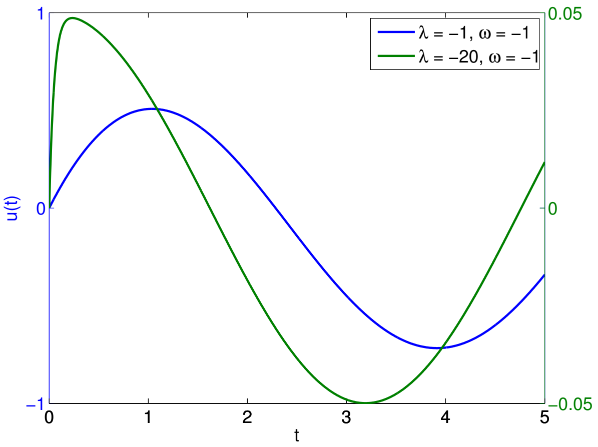


Figure 21.3: Solutions to the ODE with sinusoidal forcing.

Discretization

In order to solve an IVP numerically, we first discretize the time domain $]0, t_f]$ into J segments. The discrete time points are given by

$$t^j = j\Delta t, \quad j = 0, 1, \ldots, J = t_f/\Delta t \ ,$$

where Δt is the time step. For simplicity, we assume in this chapter that the time step is constant throughout the time integration.

The Euler Backward method is obtained by applying the first-order Backward Difference Formula (see Unit I) to the time derivative. Namely, we approximate the time derivative by

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

where $\tilde{u}^j = \tilde{u}(t^j)$ is the approximation to $u(t^j)$ and $\Delta t = t^j - t^{j-1}$ is the time step. Substituting the approximation into the differential equation, we obtain a difference equation

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

$$\tilde{u}^0 = u_0 \ ,$$

for $\tilde{u}^j$, $j = 0, \ldots, J$. Note the scheme is called "implicit" because time level j appears on the right-hand side. We can think of Euler Backward as a kind of rectangle, right integration rule — but now the integrand is not known *a priori*.

We anticipate that the solution $\tilde{u}^j$, $j = 1, \ldots, J$, approaches the true solution $u(t^j)$, $j = 1, \ldots, J$, as the time step gets smaller and the finite difference approximation approaches the continuous system. In order for this convergence to the true solution to take place, the discretization must possess two important properties: consistency and stability. Note our analysis here is more subtle than the analysis in Unit I. In Unit I we looked at the error in the finite difference approximation; here, we are interested in the error *induced* by the finite difference approximation on the approximate solution of the ODE IVP.

Consistency

Consistency is a property of a discretization that ensures that the discrete equation approximates the same process as the underlying ODE as the time step goes to zero. This is an important property, because if the scheme is not consistent with the ODE, then the scheme is modeling a different process and the solution would not converge to the true solution.

Let us define the notion of consistency more formally. We first define the *truncation error* by substituting the true solution $u(t)$ into the Euler Backward discretization, i.e.

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

Note that the truncation error, τ_{trunc}^j, measures the extent to which the exact solution to the ODE does not satisfy the difference equation. In general, the exact solution does not satisfy the difference equation, so $\tau_{\text{trunc}}^j \neq 0$. In fact, as we will see shortly, if $\tau_{\text{trunc}}^j = 0$, $j = 1, \dots, J$, then $\tilde{u}^j = u(t^j)$, i.e., $\tilde{u}^j$ is the exact solution to the ODE at the time points.

We are particularly interested in the largest of the truncation errors, which is in a sense the largest discrepancy between the differential equation and the difference equation. We denote this using the infinity norm,

$$\|\tau_{\text{trunc}}\|_\infty = \max_{j=1,\dots,J} |\tau_{\text{trunc}}^j| \ .$$

A scheme is *consistent* with the ODE if

$$\|\tau_{\text{trunc}}\|_\infty \to 0 \quad \text{as} \quad \Delta t \to 0 \ .$$

The difference equation for a consistent scheme approaches the differential equation as $\Delta t \to 0$. However, this does not necessary imply that the solution to the difference equation, $\tilde{u}(t^j)$, approaches the solution to the differential equation, $u(t^j)$.

The Euler Backward scheme is consistent. In particular

$$\|\tau_{\text{trunc}}\|_\infty \leq \frac{\Delta t}{2} \max_{t \in [0, t_f]} \left| \frac{d^2 u}{dt^2}(t) \right| \to 0 \quad \text{as} \quad \Delta t \to 0 \ .$$

We demonstrate this result below.

Begin Advanced Material

Let us now analyze the consistency of the Euler Backward integration scheme. We first apply Taylor expansion to $u(t^{j-1})$ about t^j, i.e.

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

This result is simple to derive. By the fundamental theorem of calculus,

$$\int_{t^{j-1}}^{\tau} \frac{du^2}{dt^2}(\gamma) d\gamma = \frac{du}{dt}(\tau) - \frac{du}{dt}(t^{j-1}) \ .$$

Integrating both sides over $]t^{j-1}, t^j[$,

$$\int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \frac{du^2}{dt^2}(\gamma) d\gamma \right) d\tau = \int_{t^{j-1}}^{t^j} \left(\frac{du}{dt}(\tau) \right) d\tau - \int_{t^{j-1}}^{t^j} \left(\frac{du}{dt}(t^{j-1}) \right) d\tau$$

$$= u(t^j) - u(t^{j-1}) - (t^j - t^{j-1}) \frac{du}{dt}(t^{j-1})$$

$$= u(t^j) - u(t^{j-1}) - \Delta t \frac{du}{dt}(t^{j-1}) \ .$$

Substitution of the expression to the right-hand side of the Taylor series expansion yields

$$u(t^j) - \Delta t \frac{du}{dt}(t^j) - s^j(u) = u(t^j) - \Delta t \frac{du}{dt}(t^j) - u(t^j) + u(t^{j-1}) + \Delta t \frac{du}{dt}(t^{j-1}) = u(t^{j-1}) \ ,$$

which proves the desired result.

Substituting the Taylor expansion into the expression for the truncation error,

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

$$= \frac{1}{\Delta t} \left(u(t^j) - \left(u(t^j) - \Delta t \frac{du}{dt}(t^j) - s^j(u) \right) \right) - \lambda u(t^j) - f(t^j)$$

$$= \underbrace{\frac{du}{dt}(t^j) - \lambda u(t^j) - f(t^j)}_{=0\,:\,\text{by ODE}} + \frac{s^j(u)}{\Delta t}$$

$$= \frac{s^j(u)}{\Delta t} \ .$$

We now bound the remainder term $s^j(u)$ as a function of Δt. Note that

$$s^j(u) = \int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \frac{d^2u}{dt^2}(\gamma) d\gamma \right) d\tau \le \int_{t^{j-1}}^{t^j} \left(\int_{t^{j-1}}^{\tau} \left| \frac{d^2u}{dt^2}(\gamma) \right| d\gamma \right) d\tau$$

$$\le \max_{t \in [t^{j-1}, t^j]} \left| \frac{d^2u}{dt^2}(t) \right| \int_{t^{j-1}}^{t^j} \int_{t^{j-1}}^{\tau} d\gamma d\tau$$

$$= \max_{t \in [t^{j-1}, t^j]} \left| \frac{d^2u}{dt^2}(t) \right| \frac{\Delta t^2}{2}, \quad j = 1, \ldots, J \ .$$

So, the maximum truncation error is

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

We see that

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

Thus, the Euler Backward scheme is consistent.

End Advanced Material

Stability

Stability is a property of a discretization that ensures that the error in the numerical approximation does not grow with time. This is an important property, because it ensures that a small truncation error introduced at each time step does not cause a catastrophic divergence in the solution over time.

To study stability, let us consider a homogeneous IVP,

$$\frac{du}{dt} = \lambda u \ ,$$

$$u(0) = 1 \ .$$

Recall that the true solution is of the form $u(t) = e^{\lambda t}$ and decays for $\lambda < 0$. Applying the Euler Backward scheme, we obtain

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

$$u^0 = 1 \ .$$

A scheme is said to be absolutely stable if

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

Alternatively, we can define the amplification factor, γ, as

$$\gamma \equiv \frac{|\tilde{u}^j|}{|\tilde{u}^{j-1}|} \ .$$

Absolute stability requires that $\gamma \leq 1$ for all $j = 1, \ldots, J$.

Let us now show that the Euler Backward scheme is stable for all Δt (for $\lambda < 0$). Rearranging the difference equation,

$$\tilde{u}^j - \tilde{u}^{j-1} = \lambda \Delta t \, \tilde{u}^j$$

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

$$|\tilde{u}^j| \, |1 - \lambda \Delta t| = |\tilde{u}^{j-1}| \ .$$

So, we have

$$\gamma = \frac{|\tilde{u}^j|}{|\tilde{u}^{j-1}|} = \frac{1}{|1 - \lambda \Delta t|} \ .$$

Recalling that $\lambda < 0$ (and $\Delta t > 0$), we have

$$\gamma = \frac{1}{1 - \lambda \Delta t} < 1 \ .$$

Thus, the Euler Backward scheme is stable for all Δt for the model problem considered. The scheme is said to be *unconditionally stable* because it is stable for all Δt. Some schemes are only *conditionally stable*, which means the scheme is stable for $\Delta t \leq \Delta t_{\mathrm{cr}}$, where Δt_{cr} is some critical time step.

Convergence: Dahlquist Equivalence Theorem

Now we define the notion of convergence. A scheme is convergent if the numerical approximation approaches the analytical solution as the time step is reduced. Formally, this means that

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

Note that fixed time t^j means that the time index must go to infinity (i.e., an infinite number of time steps are required) as $\Delta t \to 0$ because $t^j = j\Delta t$. Thus, convergence requires that not too much error is accumulated at each time step. Furthermore, the error generated at a given step should not grow over time.

The relationship between consistency, stability, and convergence is summarized in the Dahlquist equivalence theorem. The theorem states that consistency and stability are the necessary and sufficient condition for a convergent scheme, i.e.

$$\text{consistency} + \text{stability} \Leftrightarrow \text{convergence} \ .$$

Thus, we only need to show that a scheme is consistent and (absolutely) stable to show that the scheme is convergent. In particular, the Euler Backward scheme is convergent because it is consistent and (absolutely) stable.

Begin Advanced Material

Example 21.1.1 Consistency, stability, and convergence for Euler Backward
In this example, we will study in detail the relationship among consistency, stability, and convergence for the Euler Backward scheme. Let us denote the error in the solution 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

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

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

subtracting these two equations and using the definition of the error we obtain

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

or, rearranging the equation,

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

We see that the error itself satisfies the Euler Backward difference equation with the truncation error as the source term. Clearly, if the truncation error τ^j_{trunc} is zero for all time steps (and initial error is zero), then the error remains zero. In other words, if the truncation error is zero then the scheme produces the exact solution at each time step.

However, in general, the truncation error is nonzero, and we would like to analyze its influence on the error. Let us multiply the equation by $(1 - \lambda \Delta t)^{j-1}$ to get

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

Now, let us compute the sum for $j = 1, \ldots, n$, for some $n \le J$,

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

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

$$(1 - \lambda\Delta t)^n e^n - (1 - \lambda\Delta t)^{n-1} e^{n-1} = (1 - \lambda\Delta t)^{n-1} \Delta t \tau_{\text{trunc}}^{n}$$

$$(1 - \lambda\Delta t)^{n-1} e^{n-1} - (1 - \lambda\Delta t)^{n-2} e^{n-2} = (1 - \lambda\Delta t)^{n-2} \Delta t \tau_{\text{trunc}}^{n-1}$$

$$\vdots$$

$$(1 - \lambda\Delta t)^2 e^2 - (1 - \lambda\Delta t)^1 e^1 = (1 - \lambda\Delta t)^1 \Delta t \tau_{\text{trunc}}^{2}$$

$$(1 - \lambda\Delta t)^1 e^1 - (1 - \lambda\Delta t)^0 e^0 = (1 - \lambda\Delta t)^0 \Delta t \tau_{\text{trunc}}^{1}$$

simplifies to

$$(1 - \lambda\Delta t)^n e^n - e^0 = \sum_{j=1}^{n} (1 - \lambda\Delta t)^{j-1} \Delta t \tau_{\text{trunc}}^{j} .$$

Recall that we set $\tilde{u}^0 = \tilde{u}(t^0) = u(t^0)$, so the initial error is zero ($e^0 = 0$). Thus, we are left with

$$(1 - \lambda\Delta t)^n e^n = \sum_{j=1}^{n} (1 - \lambda\Delta t)^{j-1} \Delta t \tau_{\text{trunc}}^{j}$$

or, equivalently,

$$e^n = \sum_{j=1}^{n} (1 - \lambda\Delta t)^{j-n-1} \Delta t \tau_{\text{trunc}}^{j} .$$

Recalling that $\|\tau_{\text{trunc}}\|_\infty = \max_{j=1,\ldots,J} |\tau_{\text{trunc}}^{j}|$, we can bound the error by

$$|e^n| \le \Delta t \|\tau_{\text{trunc}}\|_\infty \sum_{j=1}^{n} (1 - \lambda\Delta t)^{j-n-1} .$$

Recalling the amplification factor for the Euler Backward scheme, $\gamma = 1/(1 - \lambda\Delta t)$, the summation can be rewritten as

$$\sum_{j=1}^{n} (1 - \lambda\Delta t)^{j-n-1} = \frac{1}{(1 - \lambda\Delta t)^n} + \frac{1}{(1 - \lambda\Delta t)^{n-1}} + \cdots + \frac{1}{(1 - \lambda\Delta t)}$$

$$= \gamma^n + \gamma^{n-1} + \cdots + \gamma .$$

Because the scheme is stable, the amplification factor satisfies $\gamma \le 1$. Thus, the sum is bounded by

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

Thus, we have

$$|e^n| \leq (n\Delta t)\|\tau_{\text{trunc}}\|_\infty = t^n\|\tau_{\text{trunc}}\|_\infty \ .$$

Furthermore, because the scheme is consistent, $\|\tau_{\text{trunc}}\|_\infty \to 0$ as $\Delta t \to 0$. Thus,

$$\|e^n\| \leq t^n\|\tau_{\text{trunc}}\|_\infty \to 0 \quad \text{as} \quad \Delta t \to 0$$

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

·

End Advanced Material

Order of Accuracy

The Dahlquist equivalence theorem shows that if a scheme is consistent and stable, then it is convergent. However, the theorem does not state how quickly the scheme converges to the true solution as the time step is reduced. Formally, a scheme is said to be p^{th}-order accurate if

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

The Euler Backward scheme is first-order accurate ($p = 1$), because

$$\|e^j\| \leq t^j\|\tau_{\text{trunc}}\|_\infty \leq t^j \frac{\Delta t}{2} \max_{t\in[0,t_f]} \left|\frac{d^2u}{dt^2}(t)\right| \leq C\Delta t^1$$

with

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

(We use here $t^j \leq t_f$.)

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

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

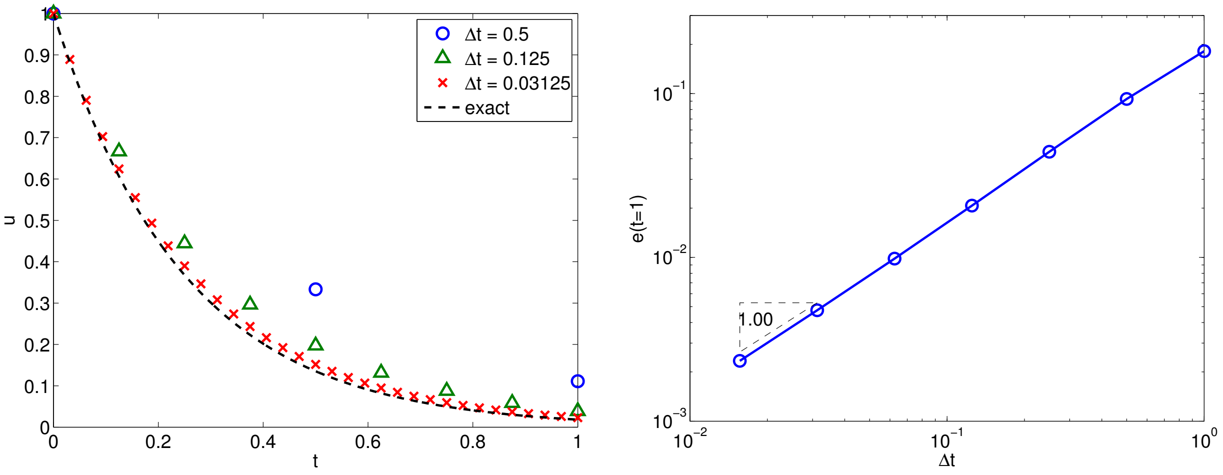
In other words, once we prove the stability of a scheme, then we just need to analyze its truncation error to understand its convergence rate. This requires little more work than checking for consistency. It is significantly simpler than deriving the expression for the evolution of the error and analyzing the error behavior directly.

Figure 21.4 shows the error convergence behavior of the Euler Backward scheme applied to the homogeneous ODE with $\lambda = -4$. The error is measured at $t = 1$. Consistent with the theory, the scheme converges at the rate of $p = 1$.

21.1.4 An Explicit Scheme: Euler Forward

Let us now introduce a new scheme, the Euler Forward scheme. The Euler Forward scheme is obtained by applying the first-order forward difference formula to the time derivative, i.e.

$$\frac{du}{dt} \approx \frac{\tilde{u}^{j+1} - \tilde{u}^j}{\Delta t} \ .$$



(a) solution

(b) error

Figure 21.4: The error convergence behavior for the Euler Backward scheme applied to the homogeneous ODE ($\lambda = -4$). Note $e(t = 1) = |u(t^j) - \tilde{u}^j|$ at $t^j = j\Delta t = 1$.

Substitution of the expression to the linear ODE yields a difference equation,

$$\frac{\tilde{u}^{j+1} - \tilde{u}^j}{\Delta t} = \lambda u^j + f(t^j), \quad j = 0, \ldots, J-1 \ ,$$

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

To maintain the same time index as the Euler Backward scheme (i.e., the difference equation involves the unknowns $\tilde{u}^j$ and $\tilde{u}^{j-1}$), let us shift the indices to obtain

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

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

The key difference from the Euler Backward scheme is that the terms on the right-hand side are evaluated at t^{j-1} instead of at t^j. Schemes for which the right-hand side does *not* involve time level j are known as "explicit" schemes. While the change may appear minor, this significantly modifies the stability. (It also changes the computational complexity, as we will discuss later.) We may view Euler Forward as a kind of "rectangle, left" integration rule.

Let us now analyze the consistency and stability of the scheme. The proof of consistency is similar to that for the Euler Backward scheme. The truncation error for the scheme is

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

To analyze the convergence of the truncation error, we apply Taylor expansion to $u(t^j)$ about t^{j-1} to obtain,

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

Thus, the truncation error simplifies to

$$\tau^j_{\text{trunc}} = \frac{1}{\Delta t}\left(u(t^{j-1}) + \Delta t \frac{du}{dt}(t^{j-1}) + s^j(u) - u(t^{j-1})\right) - \lambda u(t^{j-1}) - f(t^{j-1})$$

$$= \underbrace{\frac{du}{dt}(t^{j-1}) - \lambda u(t^{j-1}) - f(t^{j-1})}_{=0\,:\,\text{by ODE}} + \frac{s^j(u)}{\Delta t}$$

$$= \frac{s^j(u)}{\Delta t} \ .$$

In proving the consistency of the Euler Backward scheme, we have shown that $s^j(u)$ is bounded by

$$s^j(u) \le \max_{t\in[t^{j-1},t^j]} \left|\frac{d^2u}{dt^2}(t)\right| \frac{\Delta t^2}{2}, \quad j = 1,\dots,J \ .$$

Thus, the maximum truncation error is bounded by

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

Again, the truncation error converges linearly with Δt and the scheme is consistent because $\|\tau_{\text{trunc}}\|_\infty \to 0$ as $\Delta t \to 0$. Because the scheme is consistent, we only need to show that it is stable to ensure convergence.

To analyze the stability of the scheme, let us compute the amplification factor. Rearranging the difference equation for the homogeneous case,

$$\tilde{u}^j - \tilde{u}^{j-1} = \lambda \Delta t \tilde{u}^{j-1}$$

or

$$|\tilde{u}^j| = |1 + \lambda\Delta t||\tilde{u}^{j-1}|$$

which gives

$$\gamma = |1 + \lambda \Delta t| \ .$$

Thus, absolute stability (i.e., $\gamma \le 1$) requires

$$-1 \le 1 + \lambda\Delta t \le 1$$

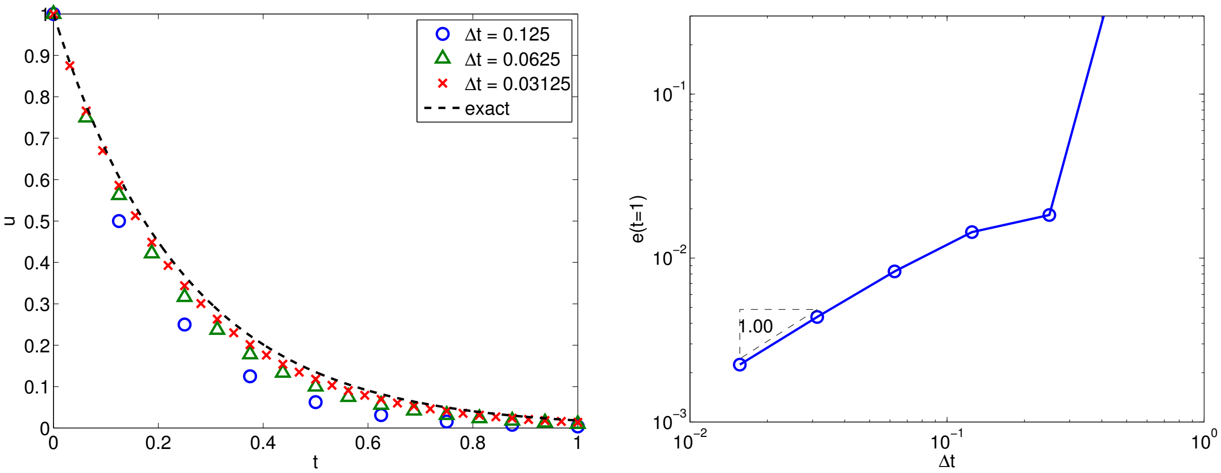
$$-2 \le \lambda\Delta t \le 0 \ .$$

Noting $\lambda\Delta t \le 0$ is a trivial condition for $\lambda < 0$, the condition for stability is

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

Note that the Euler Forward scheme is stable only for $\Delta t \le 2/|\lambda|$. Thus, the scheme is conditionally stable. Recalling the stability is a necessary condition for convergence, we conclude that the scheme converges for $\Delta t \le \Delta t_{\text{cr}}$, but diverges (i.e., blows up) with j if $\Delta t > \Delta t_{\text{cr}}$.

Figure 21.5 shows the error convergence behavior of the Euler Forward scheme applied to the homogeneous ODE with $\lambda = -4$. The error is measured at $t = 1$. The critical time step for stability is $\Delta t_{\text{cr}} = -2/\lambda = 1/2$. The error convergence plot shows that the error grows exponentially for



(a) solution

(b) error

Figure 21.5: The error convergence behavior for the Euler Forward scheme applied to $du/dt = -4u$. Note $e(t = 1) = |u(t^j) - \tilde{u}^j|$ at $t^j = j\Delta t = 1$.

$\Delta t > 1/2$. As Δt tends to zero, the numerical approximation converges to the exact solution, and the convergence rate (order) is $p = 1$ — consistent with the theory.

We should emphasize that the instability of the Euler Forward scheme for $\Delta t > \Delta t_{\mathrm{cr}}$ is *not* due to round-off errors and floating point representation (which involves "truncation," but not truncation of the variety discussed in this chapter). In particular, all of our arguments for instability hold in *infinite-precision arithmetic* as well as finite-precision arithmetic. The instability derives from the difference equation; the instability amplifies truncation error, which is a property of the difference equation and differential equation. Of course, an unstable difference equation will *also* amplify round-off errors, but that is an additional consideration and not the main reason for the explosion in Figure 21.5.

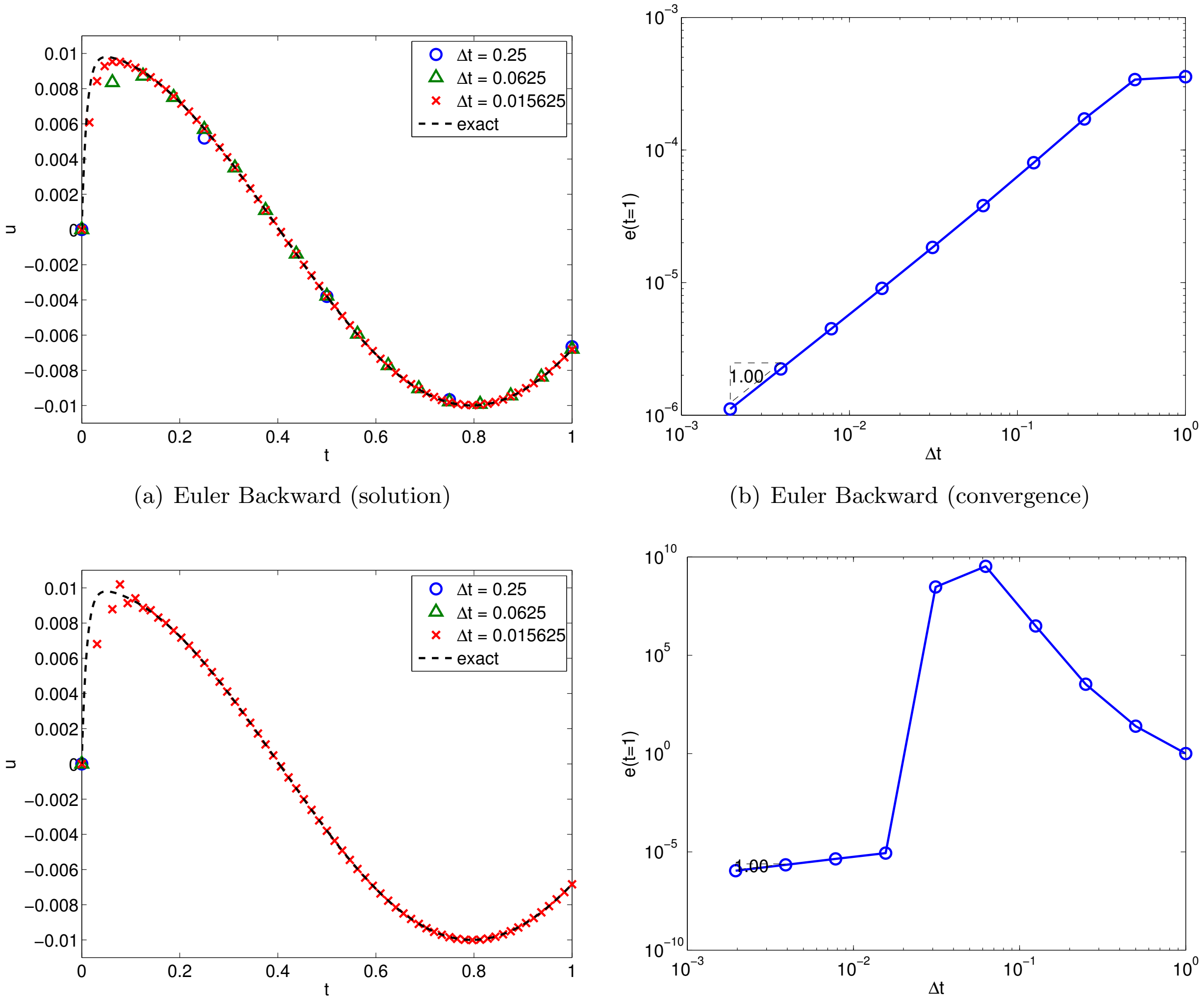
21.1.5 Stiff Equations: Implicit *vs.* Explicit

Stiff equations are the class of equations that exhibit a wide range of time scales. For example, recall the linear ODE with a sinusoidal forcing,

$$\frac{du}{dt} = \lambda t + \cos(\omega t) \ ,$$

with $|\lambda| \gg \omega$. The transient response of the solution is dictated by the time constant $1/|\lambda|$. However, this initial transient decays exponentially with time. The long time response is governed by the time constant $1/\omega \gg 1/|\lambda|$.

Let us consider the case with $\lambda = -100$ and $\omega = 4$; the time scales differ by a factor of 25. The result of applying the Euler Backward and Euler Forward schemes with several different time steps is shown in Figure 21.6. Recall that the Euler Backward scheme is stable for any time step for $\lambda < 0$. The numerical result confirms that the solution is bounded for all time steps considered. While a large time step (in particular $\Delta t > 1/|\lambda|$) results in an approximation which does not capture the initial transient, the long term behavior of the solution is still well represented. Thus, if the initial transient is not of interest, we can use a Δt optimized to resolve only the long term behavior associated with the characteristic time scale of $1/\omega$ — in other words, $\Delta t \sim O(1/10)$,



(a) Euler Backward (solution)

(b) Euler Backward (convergence)

(c) Euler Forward (solution)

(d) Euler Forward (convergence)

Figure 21.6: Application of the Euler Backward and Euler Forward schemes to a stiff equation. Note $e(t = 1) = |u(t^j) - \tilde{u}^j|$ at $t^j = j\Delta t = 1$.

rather than $\Delta t \sim O(1/|\lambda|)$. If $|\lambda| \gg \omega$, then we significantly reduce the number of time steps (and thus the computational cost).

Unlike its implicit counterpart, the Euler Forward method is only conditionally stable. In particular, the critical time step for this problem is $\Delta t_{\mathrm{cr}} = 2/|\lambda| = 0.02$. Thus, even if we are not interested in the initial transient, we cannot use a large time step because the scheme would be unstable. Only one of the three numerical solution ($\Delta t = 1/64 < \Delta t_{\mathrm{cr}}$) is shown in Figure 21.6(c) because the other two time steps ($\Delta t = 1/16$, $\Delta t = 1/4$) result in an unstable discretization and a useless approximation. The exponential growth of the error for $\Delta t > \Delta t_{\mathrm{cr}}$ is clearly reflected in Figure 21.6(d).

Stiff equations are ubiquitous in the science and engineering context; in fact, it is not uncommon to see scales that differ by over ten orders of magnitude. For example, the time scale associated with the dynamics of a passenger jet is several orders of magnitude larger than the time scale associated with turbulent eddies. If the dynamics of the smallest time scale is not of interest, then an unconditionally stable scheme that allows us to take arbitrarily large time steps may be computationally advantageous. In particular, we can select the time step that is necessary to achieve sufficient accuracy without any time step restriction arising from the stability consideration. Put another way, integration of a stiff system using a conditionally stable method may place a stringent requirement on the time step, rendering the integration prohibitively expensive. As none of the explicit schemes are unconditionally stable, implicit schemes are often preferred for stiff equations.

We might conclude from the above that explicit schemes serve very little purpose. In fact, this is not the case, because the story is a bit more complicated. 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, $1+(\lambda\Delta t)$. When we consider real problems of interest — systems, often large systems, of many and often nonlinear ODEs — these scalar algebraic operations of division for implicit schemes and multiplication for explicit schemes will translate into matrix inversion (more precisely, solution of matrix equations) and matrix multiplication, respectively. In general, and as we shall see in Unit V, matrix inversion is much more costly than matrix multiplication.

Hence the total cost equation is more nuanced. An implicit scheme will typically enjoy a larger time step and hence fewer time steps — but require more work for each time step (matrix solution). In contrast, an explicit scheme may require a much smaller time step and hence many more time steps — but will entail much less work for each time step. For stiff equations in which the Δt for accuracy is much, much larger than the Δt_{cr} required for stability (of explicit schemes), typically implicit wins. On the other hand, for non-stiff equations, in which the Δt for accuracy might be on the same order as Δt_{cr} required for stability (of explicit schemes), explicit can often win: in such cases we would in any event (for reasons of accuracy) choose a $\Delta t \approx \Delta t_{\text{cr}}$; hence, since an explicit scheme will be stable for this Δt, we might as well choose an explicit scheme to minimize the work per time step.

Begin Advanced Material

21.1.6 Unstable Equations

End Advanced Material

21.1.7 Absolute Stability and Stability Diagrams

We have learned that different integration schemes exhibit different stability characteristics. In particular, implicit methods tend to be more stable than explicit methods. To characterize the stability of different numerical integrators, let us introduce absolute *stability diagrams*. These diagrams allow us to quickly analyze whether an integration scheme will be stable for a given system.

Euler Backward

Let us construct the stability diagram for the Euler Backward scheme. We start with the homogeneous equation

$$\frac{dz}{dt} = \lambda z \ .$$

So far, we have only considered a real λ; now we allow λ to be a general complex number. (Later λ will represent an eigenvalue of a system, which in general will be a complex number.) The Euler

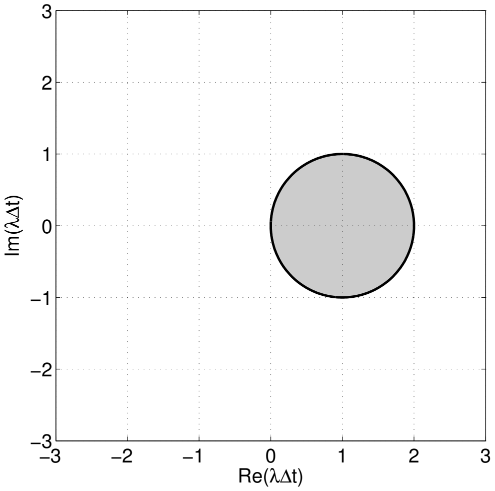


Figure 21.7: The absolute stability diagram for the Euler Backward scheme.

Backward discretization of the equation is

$$\frac{\tilde{z}^j - \tilde{z}^{j-1}}{\Delta t} = \lambda \tilde{z}^j \quad \Rightarrow \quad \tilde{z}^j = (1 - (\lambda \Delta t))^{-1} \tilde{z}^{j-1} \ .$$

Recall that we defined the absolute stability as the region in which the amplification factor $\gamma \equiv |\tilde{z}^j|/|\tilde{z}^{j-1}|$ is less than or equal to unity. This requires

$$\gamma = \frac{|\tilde{z}^j|}{|\tilde{z}^{j-1}|} = \left| \frac{1}{1 - (\lambda \Delta t)} \right| \leq 1 \ .$$

We wish to find the values of $(\lambda \Delta t)$ for which the numerical solution exhibits a stable behavior (i.e., $\gamma \leq 1$). A simple approach to achieve this is to solve for the stability *boundary* by setting the amplification factor to $1 = |e^{i\theta}|$, i.e.

$$e^{i\theta} = \frac{1}{1 - (\lambda \Delta t)} \ .$$

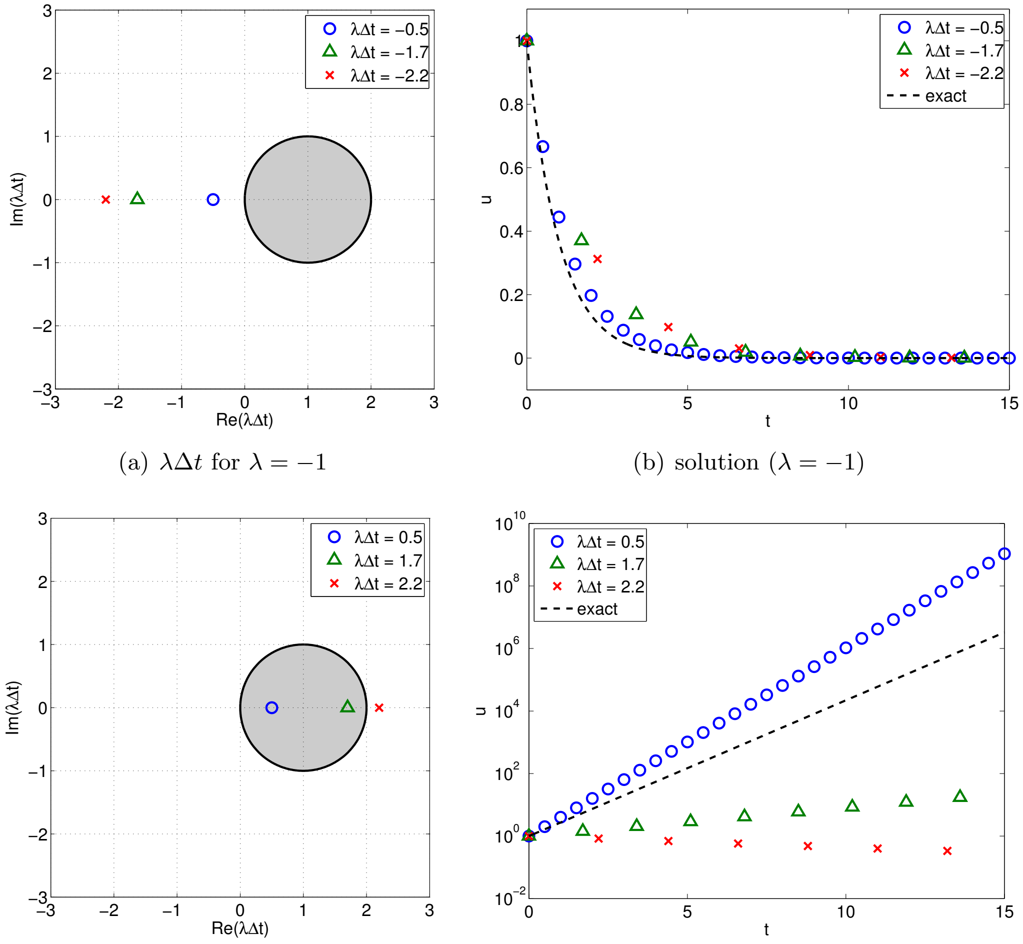
Solving for $(\lambda \Delta t)$, we obtain

$$(\lambda \Delta t) = 1 - e^{-i\theta} \ .$$

Thus, the stability boundary for the Euler Backward scheme is a circle of unit radius (the "one" multiplying $e^{i\theta}$) centered at 1 (the one directly after the = sign).

To deduce on which side of the boundary the scheme is stable, we can check the amplification factor evaluated at a point not on the circle. For example, if we pick $\lambda \Delta t = -1$, we observe that $\gamma = 1/2 \leq 1$. Thus, the scheme is stable *outside* of the unit circle. Figure 21.7 shows the stability diagram for the Euler Backward scheme. The scheme is *unstable* in the shaded region; it is *stable* in the unshaded region; it is *neutrally* stable, $|\tilde{z}^j| = |\tilde{z}^{j-1}|$, *on* the unit circle. The unshaded region ($\gamma < 1$) and the boundary of the shaded and unshaded regions ($\gamma = 1$) represent the absolute stability region; the entire picture is denoted the absolute stability diagram.

To gain understanding of the stability diagram, let us consider the behavior of the Euler Backward scheme for a few select values of $\lambda \Delta t$. First, we consider a stable homogeneous equation, with $\lambda = -1 < 0$. We consider three different values of $\lambda \Delta t$, -0.5, -1.7, and -2.2. Figure 21.8(a) shows



(a) $\lambda\Delta t$ for $\lambda = -1$

(b) solution ($\lambda = -1$)

(c) $\lambda\Delta t$ for $\lambda = 1$

(d) solution ($\lambda = 1$)

Figure 21.8: The behavior of the Euler Backward scheme for selected values of $(\lambda\Delta t)$.

the three points on the stability diagram that correspond to these choices of $\lambda\Delta t$. All three points lie in the unshaded region, which is a stable region. Figure 21.8(b) shows that all three numerical solutions decay with time as expected. While the smaller Δt results in a smaller error, all schemes are stable and converge to the same steady state solution.

Begin Advanced Material

Next, we consider an unstable homogeneous equation, with $\lambda = 1 > 0$. We again consider three different values of $\lambda\Delta t$, 0.5, 1.7, and 2.2. Figure 21.8(c) shows that two of these points lie in the unstable region, while $\lambda\Delta t = 2.2$ lies in the stable region. Figure 21.8(d) confirms that the solutions for $\lambda\Delta t = 0.5$ and 1.7 grow with time, while $\lambda\Delta t = 2.2$ results in a decaying solution. The true solution, of course, grows exponentially with time. Thus, if the time step is too large (specifically $\lambda\Delta t > 2$), then the Euler Backward scheme can produce a decaying solution even if the true solution grows with time — which is undesirable; nevertheless, as $\Delta t \to 0$, we obtain the correct behavior. In general, the interior of the absolute stability region should not include $\lambda\Delta t = 0$. (In fact $\lambda\Delta t = 0$ should be on the stability boundary.)

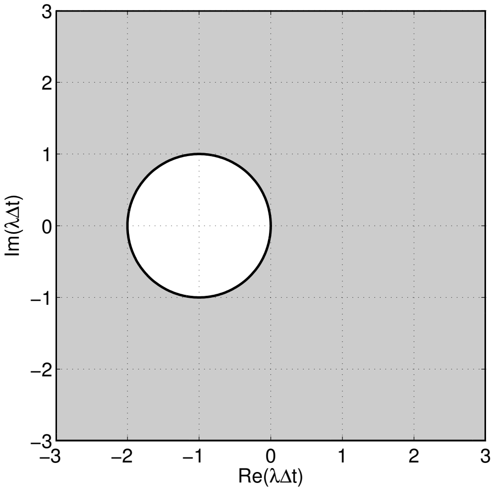


Figure 21.9: The absolute stability diagram for the Euler Forward scheme. The white area corresponds to stability (the absolute stability region) and the gray area to instability.

End Advanced Material

Euler Forward

Let us now analyze the absolute stability characteristics of the Euler Forward scheme. Similar to the Euler Backward scheme, we start with the homogeneous equation. The Euler Forward discretization of the equation yields

$$\frac{\tilde{z}^j - \tilde{z}^{j-1}}{\Delta t} = \lambda \tilde{z}^{j-1} \quad \Rightarrow \quad \tilde{z}^j = (1 + (\lambda \Delta t))\tilde{z}^{j-1} \ .$$

The stability boundary, on which the amplification factor is unity, is given by

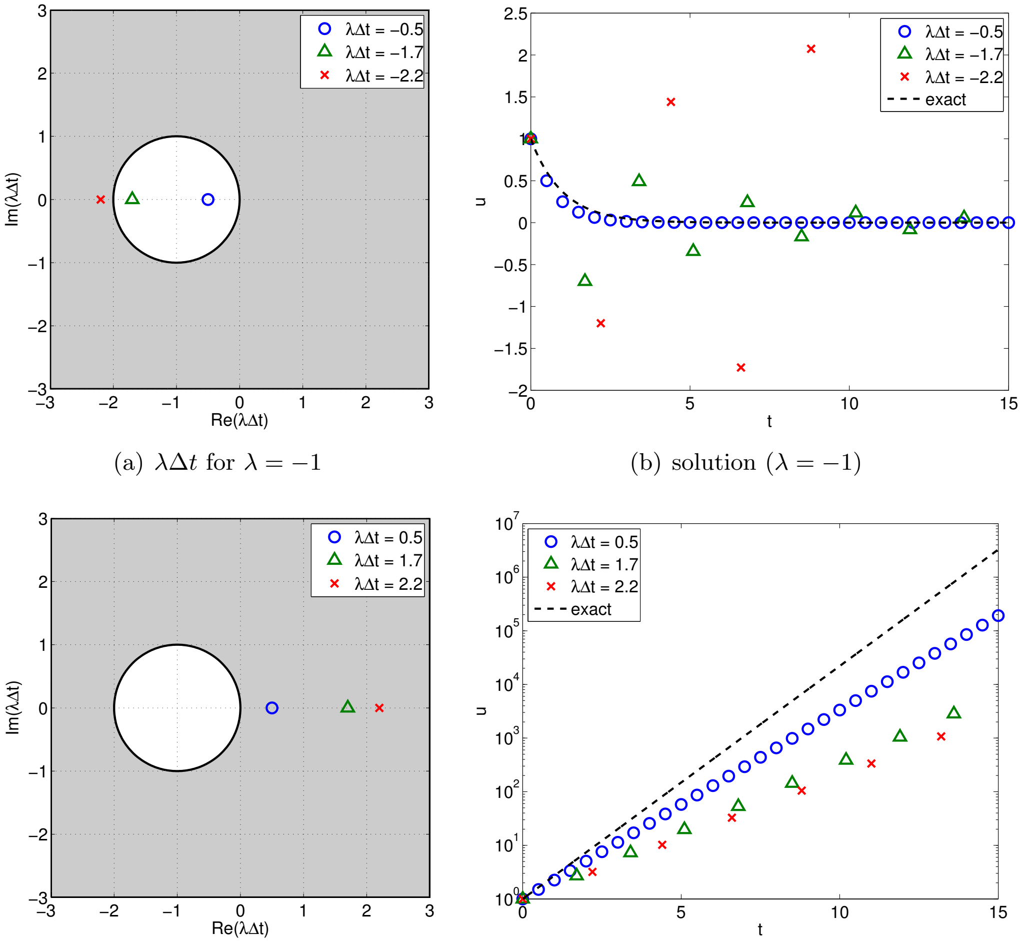
$$\gamma = |1 + (\lambda \Delta t)| = 1 \quad \Rightarrow \quad (\lambda \Delta t) = e^{-i\theta} - 1 \ .$$

The stability boundary is a circle of unit radius centered at -1. Substitution of, for example, $\lambda\Delta t = -1/2$, yields $\gamma(\lambda\Delta t = -1/2) = 1/2$, so the amplification is less than unity inside the circle. The stability diagram for the Euler Forward scheme is shown in Figure 21.9.

As in the Euler Backward case, let us pick a few select values of $\lambda\Delta t$ and study the behavior of the Euler Forward scheme. The stability diagram and solution behavior for a stable ODE ($\lambda = -1 < 0$) are shown in Figure 21.10(a) and 21.10(b), respectively. The cases with $\lambda\Delta t = -0.5$ and -1.7 lie in the stable region of the stability diagram, while $\lambda\Delta t = -2.2$ lies in the unstable region. Due to instability, the numerical solution for $\lambda\Delta t = -2.2$ diverges exponentially with time, even though the true solution decays with time. The solution for $\lambda\Delta t = -1.7$ shows some oscillation, but the magnitude of the oscillation decays with time, agreeing with the stability diagram. (For an unstable ODE ($\lambda = 1 > 0$), Figure 21.10(c) shows that all time steps considered lie in the unstable region of the stability diagram. Figure 21.10(d) confirms that all these choices of Δt produce a growing solution.)

21.1.8 Multistep Schemes

We have so far considered two schemes: the Euler Backward scheme and the Euler Forward scheme. These two schemes compute the state $\tilde{u}^j$ from the previous state $\tilde{u}^{j-1}$ and the source function



(a) $\lambda\Delta t$ for $\lambda = -1$

(b) solution ($\lambda = -1$)

(c) $\lambda\Delta t$ for $\lambda = 1$

(d) solution ($\lambda = 1$)

Figure 21.10: The behavior of the Euler Forward scheme for selected values of $\lambda\Delta t$.

evaluated at t^j or t^{j-1}. The two schemes are special cases of *multistep schemes*, where the solution at the current time $\tilde{u}^j$ is approximated from the previous solutions. In general, for an ODE of the form

$$\frac{du}{dt} = g(u,t) \ ,$$

a K-step multistep scheme takes the form

$$\sum_{k=0}^{K} \alpha_k \tilde{u}^{j-k} = \Delta t \sum_{k=0}^{K} \beta_k g^{j-k}, \quad j = 1, \dots, J \ ,$$

$$\tilde{u}^j = u_0 \ ,$$

where $g^{j-k} = g(\tilde{u}^{j-k}, t^{j-k})$. Note that the linear ODE we have been considering results from the choice $g(u,t) = \lambda u + f(t)$. A K-step multistep scheme requires solutions (and derivatives) at K previous time steps. Without loss of generality, we choose $\alpha_0 = 1$. A scheme is uniquely defined by choosing $2K + 1$ coefficients, α_k, $k = 1, \dots, K$, and β_k, $k = 0, \dots, K$.

Multistep schemes can be categorized into implicit and explicit schemes. If we choose $\beta_0 = 0$, then $\tilde{u}^j$ does not appear on the right-hand side, resulting in an explicit scheme. As discussed before, explicit schemes are only conditionally stable, but are computationally less expensive per step. If we choose $\beta_0 = 0$, then $\tilde{u}^j$ appears on the right-hand side, resulting in an implicit scheme. Implicit schemes tend to be more stable, but are more computationally expensive per step, especially for a system of nonlinear ODEs.

Let us recast the Euler Backward and Euler Forward schemes in the multistep method framework.

Example 21.1.2 Euler Backward as a multistep scheme
The Euler Backward scheme is a 1-step method with the choices

$$\alpha_1 = -1, \quad \beta_0 = 1, \quad \text{and} \quad \beta_1 = 0 \ .$$

This results in

$$\tilde{u}^j - \tilde{u}^{j-1} = \Delta t g^j, \quad j = 1, \dots, J \ .$$

Example 21.1.3 Euler Forward as a multistep scheme
The Euler Forward scheme is a 1-step method with the choices

$$\alpha_1 = -1, \quad \beta_0 = 0, \quad \text{and} \quad \beta_1 = 1 \ .$$

This results in

$$\tilde{u}^j - \tilde{u}^{j-1} = \Delta t g^{j-1}, \quad j = 1, \dots, J \ .$$

Now we consider three families of multistep schemes: Adams-Bashforth, Adams-Moulton, and Backward Differentiation Formulas.

Adams-Bashforth Schemes

Adams-Bashforth schemes are explicit multistep time integration schemes ($\beta_0 = 0$). Furthermore, we restrict ourselves to

$$\alpha_1 = -1 \quad \text{and} \quad \alpha_k = 0, \quad k = 2, \dots, K \ .$$

The resulting family of the schemes takes the form

$$\tilde{u}^j = \tilde{u}^{j-1} + \sum_{k=1}^{K} \beta_k g^{j-k} \ .$$

Now we must choose β_k, $k = 1, \dots K$, to define a scheme. To choose the appropriate values of β_k, we first note that the true solution $u(t^j)$ and $u(t^{j-1})$ are related by

$$u(t^j) = u(t^{j-1}) + \int_{t^{j-1}}^{t^j} \frac{du}{dt}(\tau) d\tau = u(t^{j-1}) + \int_{t^{j-1}}^{t^j} g(u(\tau), \tau) d\tau \ . \tag{21.1}$$

Then, we approximate the integrand $g(u(\tau), \tau)$, $\tau \in (t^{j-1}, t^j)$, using the values g^{j-k}, $k = 1, \dots, K$. Specifically, we construct a $(K-1)^{\text{th}}$-degree polynomial $p(\tau)$ using the K data points, i.e.

$$p(\tau) = \sum_{k=1}^{K} \phi_k(\tau) g^{j-k} \ ,$$

where $\phi_k(\tau)$, $k = 1, \dots, K$, are the Lagrange interpolation polynomials defined by the points t^{j-k}, $k = 1, \dots, K$. Recalling the polynomial interpolation theory from Unit I, we note that the $(K-1)^{\text{th}}$-degree polynomial interpolant is K^{th}-order accurate for $g(u(\tau), \tau)$ sufficiently smooth, i.e.

$$p(\tau) = g(u(\tau), \tau) + \mathcal{O}(\Delta t^K) \ .$$

(Note in fact here we consider "extrapolation" of our interpolant.) Thus, we expect the order of approximation to improve as we incorporate more points given sufficient smoothness. Substitution of the polynomial approximation of the derivative to Eq. (21.1) yields

$$u(t^j) \approx u(t^{j-1}) + \int_{t^{j-1}}^{t^j} \sum_{k=1}^{K} \phi_k(\tau) g^{j-k} d\tau = u(t^{j-1}) + \sum_{k=1}^{K} \int_{t^{j-1}}^{t^j} \phi_k(\tau) d\tau \, g^{j-k} \ .$$

To simplify the integral, let us consider the change of variable $\tau = t^j - (t^j - t^{j-1})\hat{\tau} = t^j - \Delta t \hat{\tau}$. The change of variable yields

$$u(t^j) \approx u(t^{j-1}) + \Delta t \sum_{k=1}^{K} \int_0^1 \hat{\phi}_k(\hat{\tau}) d\hat{\tau} \, g^{j-k} \ ,$$

where the $\hat{\phi}_k$ are the Lagrange polynomials associated with the interpolation points $\hat{\tau} = 1, 2, \dots, K$. We recognize that the approximation fits the Adams-Bashforth form if we choose

$$\beta_k = \int_0^1 \hat{\phi}_k(\hat{\tau}) d\hat{\tau} \ .$$

Let us develop a few examples of Adams-Bashforth schemes.

Example 21.1.4 1-step Adams-Bashforth (Euler Forward)
The 1-step Adams-Bashforth scheme requires evaluation of β_1. The Lagrange polynomial for this case is a constant polynomial, $\hat{\phi}_1(\hat{\tau}) = 1$. Thus, we obtain

$$\beta_1 = \int_0^1 \hat{\phi}_1(\hat{\tau}) d\hat{\tau} = \int_0^1 1 d\hat{\tau} = 1 \ .$$

Thus, the scheme is

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t g^{j-1} \ ,$$

which is the Euler Forward scheme, first-order accurate.

·

Example 21.1.5 2-step Adams-Bashforth
The 2-step Adams-Bashforth scheme requires specification of β_1 and β_2. The Lagrange interpolation polynomials for this case are linear polynomials

$$\hat{\phi}_1(\hat{\tau}) = -\hat{\tau} + 2 \quad \text{and} \quad \hat{\phi}_2(\hat{\tau}) = \hat{\tau} - 1 \ .$$

It is easy to verify that these are the Lagrange polynomials because $\hat{\phi}_1(1) = \hat{\phi}_2(2) = 1$ and $\hat{\phi}_1(2) = \hat{\phi}_2(1) = 0$. Integrating the polynomials

$$\beta_1 = \int_0^1 \phi_1(\hat{\tau}) d\hat{\tau} = \int_0^1 (-\hat{\tau} + 2) d\hat{\tau} = \frac{3}{2} \ ,$$

$$\beta_2 = \int_0^1 \phi_2(\hat{\tau}) d\hat{\tau} = \int_0^1 (\hat{\tau} - 1) d\hat{\tau} = -\frac{1}{2} \ .$$

The resulting scheme is

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t \left(\frac{3}{2} g^{j-1} - \frac{1}{2} g^{j-2} \right) .$$

This scheme is second-order accurate.

·

Adams-Moulton Schemes

Adams-Moulton schemes are implicit multistep time integration schemes ($\beta_0 \neq 0$). Similar to Adams-Bashforth schemes, we restrict ourselves to

$$\alpha_1 = -1 \quad \text{and} \quad \alpha_k = 0, \quad k = 2, \ldots, K \ .$$

The Adams-Moulton family of the schemes takes the form

$$\tilde{u}^j = \tilde{u}^{j-1} + \sum_{k=0}^{K} \beta_k g^{j-k} \ .$$

We must choose β_k, $k = 1, \ldots, K$ to define a scheme. The choice of β_k follows exactly the same procedure as that for Adams-Bashforth. Namely, we consider the expansion of the form Eq. (21.1) and approximate $g(u(\tau), \tau)$ by a polynomial. This time, we have $K+1$ points, thus we construct a K^{th}-degree polynomial

$$p(\tau) = \sum_{k=0}^{K} \phi_k(\tau) g^{j-k} \ ,$$

where $\phi_k(\tau)$, $k = 0, \ldots, K$, are the Lagrange interpolation polynomials defined by the points t^{j-k}, $k = 0, \ldots, K$. Note that these polynomials are different from those for the Adams-Bashforth schemes due to the inclusion of t^j as one of the interpolation points. (Hence here we consider true interpolation, not extrapolation.) Moreover, the interpolation is now $(K+1)^{\mathrm{th}}$-order accurate.

Using the same change of variable as for Adams-Bashforth schemes, $\tau = t^j - \Delta t \hat{\tau}$, we arrive at a similar expression,

$$u(t^j) \approx u(t^{j-1}) + \Delta t \sum_{k=0}^{K} \int_0^1 \hat{\phi}_k(\hat{\tau}) d\hat{\tau} g^{j-k} \ ,$$

for the Adams-Moulton schemes; here the $\hat{\phi}_k$ are the K^{th}-degree Lagrange polynomials defined by the points $\hat{\tau} = 0, 1, \ldots, K$. Thus, the β_k are given by

$$\beta_k = \int_0^1 \hat{\phi}_k(\hat{\tau}) d\hat{\tau} \ .$$

Let us develop a few examples of Adams-Moulton schemes.

Example 21.1.6 0-step Adams-Moulton (Euler Backward)
The 0-step Adams-Moulton scheme requires just one coefficient, β_0. The "Lagrange" polynomial is 0^{th} degree, i.e. a constant function $\hat{\phi}_0(\hat{\tau}) = 1$, and the integration of the constant function over the unit interval yields

$$\beta_0 = \int_0^1 \hat{\phi}_0(\hat{\tau}) d\hat{\tau} = \int_0^1 1 d\hat{\tau} = 1.$$

Thus, the 0-step Adams-Moulton scheme is given by

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t g^j,$$

which in fact is the Euler Backward scheme. Recall that the Euler Backward scheme is first-order accurate.

·

Example 21.1.7 1-step Adams-Moulton (Crank-Nicolson)
The 1-step Adams-Moulton scheme requires determination of two coefficients, β_0 and β_1. The Lagrange polynomials for this case are linear polynomials

$$\hat{\phi}_0(\hat{\tau}) = -\tau + 1 \quad \text{and} \quad \hat{\phi}_1(\hat{\tau}) = \tau \ .$$

Integrating the polynomials,

$$\beta_0 = \int_0^1 \hat{\phi}_0(\hat{\tau}) d\hat{\tau} = \int_0^1 (-\tau + 1) d\hat{\tau} = \frac{1}{2} \ ,$$

$$\beta_1 = \int_0^1 \hat{\phi}_1(\hat{\tau}) d\hat{\tau} = \int_0^1 \hat{\tau} d\hat{\tau} = \frac{1}{2} \ .$$

The choice of β_k yields the Crank-Nicolson scheme

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t \left(\frac{1}{2} g^j + \frac{1}{2} g^{j-1} \right) .$$

The Crank-Nicolson scheme is second-order accurate. We can view Crank-Nicolson as a kind of "trapezoidal" rule.

·

Example 21.1.8 2-step Adams-Moulton
The 2-step Adams-Moulton scheme requires three coefficients, β_0, β_1, and β_2. The Lagrange polynomials for this case are the quadratic polynomials

$$\hat{\phi}_0(\hat{\tau}) = \frac{1}{2}(\hat{\tau}-1)(\hat{\tau}-2) = \frac{1}{2}(\hat{\tau}^2 - 3\hat{\tau} + 2) \ ,$$

$$\hat{\phi}_1(\hat{\tau}) = -\hat{\tau}(\hat{\tau}-2) = -\hat{\tau}^2 + 2\hat{\tau} \ ,$$

$$\hat{\phi}_2(\hat{\tau}) = \frac{1}{2}\hat{\tau}(\hat{\tau}-1) = \frac{1}{2}\left(\hat{\tau}^2 - \hat{\tau}\right) .$$

Integrating the polynomials,

$$\beta_0 = \int_0^1 \hat{\phi}_0(\hat{\tau}) d\hat{\tau} = \int_0^1 \frac{1}{2}(\hat{\tau}^2 - 3\hat{\tau} + 2)\hat{\tau} = \frac{5}{12}$$

$$\beta_1 = \int_0^1 \hat{\phi}_1(\hat{\tau}) d\hat{\tau} = \int_0^1 (-\hat{\tau}^2 + 2\hat{\tau}) d\hat{\tau} = \frac{2}{3} \ ,$$

$$\beta_2 = \int_0^1 \hat{\phi}_2(\hat{\tau}) d\hat{\tau} = \int_0^1 \frac{1}{2}\left(\hat{\tau}^2 - \hat{\tau}\right) d\hat{\tau} = -\frac{1}{12} \ .$$

Thus, the 2-step Adams-Moulton scheme is given by

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t \left(\frac{5}{12} g^j + \frac{2}{3} g^{j-1} - \frac{1}{12} g^{j-2} \right) .$$

This AM2 scheme is third-order accurate.

·

Convergence of Multistep Schemes: Consistency and Stability

Let us now introduce techniques for analyzing the convergence of a multistep scheme. Due to the Dahlquist equivalence theorem, we only need to show that the scheme is consistent and stable.

To show that the scheme is consistent, we need to compute the truncation error. Recalling that the local truncation error is obtained by substituting the exact solution to the difference equation (normalized such that $\tilde{u}^j$ has the coefficient of 1) and dividing by Δt, we have for any multistep schemes

$$\tau_{\text{trunc}}^j = \frac{1}{\Delta t} \left[u(t^j) + \sum_{k=1}^{K} \alpha_k \, u(t^{j-k}) \right] - \sum_{k=0}^{K} \beta_k \, g(t^{j-k}, u(t^{j-k})) \ .$$

For simplicity we specialize our analysis to the Adams-Bashforth family, such that

$$\tau^j_{\text{trunc}} = \frac{1}{\Delta t} \quad u(t^j) - u(t^{j-1})\Big) - \sum_{k=1}^{K} \beta_k\, g(t^{j-k}, u(t^{j-k}))\ .$$

We recall that the coefficients β_k were selected to match the extrapolation from polynomial fitting. Backtracking the derivation, we simplify the sum as follows

$$\begin{aligned}\sum_{k=1}^{K} \beta_k\, g(t^{j-k}, u(t^{j-k})) &= \sum_{k=1}^{K} \int_0^1 \hat{\phi}_k(\hat{\tau}) d\hat{\tau}\; g(t^{j-k}, u(t^{j-k})) \\ &= \sum_{k=1}^{K} \frac{1}{\Delta t} \int_{t^{j-1}}^{t^j} \phi_k(\tau) d\tau\; g(t^{j-k}, u(t^{j-k})) \\ &= \frac{1}{\Delta t} \int_{t^{j-1}}^{t^j} \left[\sum_{k=1}^{K} \phi_k(\tau)\, g(t^{j-k}, u(t^{j-k})) \right] d\tau \\ &= \frac{1}{\Delta t} \int_{t^{j-1}}^{t^j} p(\tau) d\tau\ .\end{aligned}$$

We recall that $p(\tau)$ is a $(K-1)^{\text{th}}$

$^{\text{th}}$-order accurate interpolation with the error $\mathcal{O}(\Delta t^K)$. Thus,

$$\begin{aligned}\tau^j_{\text{trunc}} &= \frac{1}{\Delta t}\Big(u(t^j) - u(t^{j-1})\Big) - \sum_{k=1}^{K} \beta_k\, g(t^{j-k}, u(t^{j-k})) \\ &= \frac{1}{\Delta t}\Big(u(t^j) - u(t^{j-1})\Big) - \frac{1}{\Delta t}\int_{t^{j-1}}^{t^j} g(\tau, u(\tau)) d\tau + \frac{1}{\Delta t}\int_{j^{j-1}}^{t^j} \mathcal{O}(\Delta t^K) d\tau \\ &= \frac{1}{\Delta t}\left[u(t^j) - u(t^{j-1}) - \int_{t^{j-1}}^{t^j} g(\tau, u(\tau)) d\tau \right] + \mathcal{O}(\Delta t^K) \\ &= \mathcal{O}(\Delta t^K)\ .\end{aligned}$$

Note that the term in the bracket vanishes from $g = du/dt$ and the fundamental theorem of calculus. The truncation error of the scheme is $\mathcal{O}(\Delta t^K)$. In particular, since $K > 0$, $\tau_{\text{trunc}} \to 0$ as $\Delta t \to 0$ and the Adams-Bashforth schemes are consistent. Thus, if the schemes are stable, they would converge at Δt^K.

The analysis of stability relies on a solution technique for difference equations. We first restrict ourselves to linear equation of the form $g(t, u) = \lambda u$. By rearranging the form of difference equation for the multistep methods, we obtain

$$\sum_{k=0}^{K} (\alpha_k - (\lambda \Delta t)\, \beta_k)\, \tilde{u}^{j-k} = 0, \quad j = 1, \ldots, J\ .$$

The solution to the difference equation is governed by the initial condition and the K roots of the polynomial

$$q(x) = \sum_{k=0}^{K} (\alpha_k - (\lambda \Delta t)\, \beta_k) x^{K-k} \ .$$

In particular, for any initial condition, the solution will exhibit a stable behavior if all roots r_k, $k = 1, \ldots, K$, have magnitude less than or equal to unity. Thus, the absolute stability condition for multistep schemes is

$$(\lambda \Delta t) \text{ such that } |r_K| \leq 1, \quad k = 1, \ldots, K \ ,$$

where r_k, $k = 1, \ldots, K$ are the roots of q.

Example 21.1.9 Stability of the 2-step Adams-Bashforth scheme
Recall that the 2-step Adams-Bashforth results from the choice

$$\alpha_0 = 1, \quad \alpha_1 = -1, \quad \alpha_2 = 0, \quad \beta_0 = 0, \quad \beta_1 = \frac{3}{2}, \quad \text{and} \quad \beta_2 = -\frac{1}{2} \ .$$

The stability of the scheme is governed by the roots of the polynomial

$$q(x) = \sum_{k=0}^{2} (\alpha_k - (\lambda \Delta t)\, \beta_k) x^{2-k} = x^2 + \left(-1 - \frac{3}{2}(\lambda \Delta t) \right) x + \frac{1}{2}(\lambda \Delta t) = 0 \ .$$

The roots of the polynomial are given by

$$r_{1,2} = \frac{1}{2} \left[1 + \frac{3}{2}(\lambda \Delta t) \pm \sqrt{\left(1 + \frac{3}{2}(\lambda \Delta t) \right)^2 - 2(\lambda \Delta t)} \right] .$$

We now look for $(\lambda \Delta t)$ such that $|r_1| \leq 1$ and $|r_2| \leq 1$.

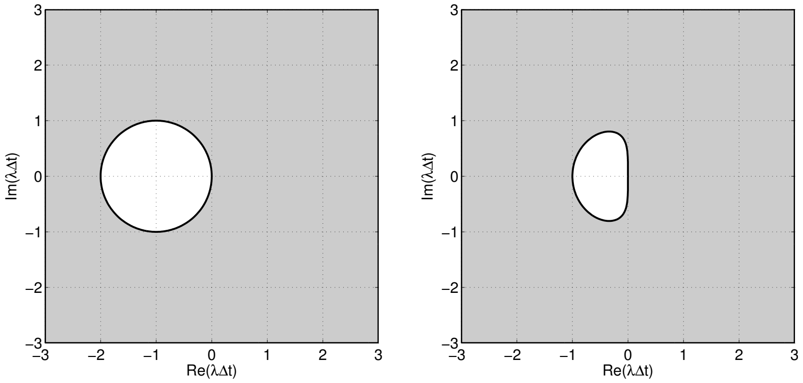
It is a simple matter to determine if a particular $\lambda \Delta t$ is inside, on the boundary of, or outside the absolute stability region. For example, for $\lambda \Delta t = -1$ we obtain $r_1 = -1$, $r_2 = 1/2$ and hence — since $|r_1| = 1$ — $\lambda \Delta t = -1$ is in fact on the boundary of the absolute stability diagram. Similarly, it is simple to confirm that $\lambda \Delta t = -1/2$ yields both r_1 and r_2 of modulus strictly less than 1, and hence $\lambda \Delta t = -1/2$ is inside the absolute stability region. We can thus in principle check each point $\lambda \Delta t$ (or enlist more sophisticated solution procedures) in order to construct the full absolute stability diagram.

We shall primarily be concerned with the *use* of the stability diagram rather than the construction of the stability diagram — which for most schemes of interest are already derived and well documented. We present in Figure 21.11(b) the absolute stability diagram for the 2-step Adams-Bashforth scheme. For comparison we show in Figure 21.11(a) the absolute stability diagram for Euler Forward, which is the 1-step Adams-Bashforth scheme. Note that the stability region of the Adams-Bashforth schemes are quite small; in fact the stability region decreases further for higher order Adams-Bashforth schemes. Thus, the method is only well suited for non-stiff equations.

———— · ————

Example 21.1.10 Stability of the Crank-Nicolson scheme
Let us analyze the absolute stability of the Crank-Nicolson scheme. Recall that the stability of a multistep scheme is governed by the roots of the polynomial

$$q(x) = \sum_{k=0}^{K} (\alpha_k - \lambda \Delta t\, \beta_k)\, x^{K-k} \ .$$



(a) Euler Forward (AB1)

(b) 2-step Adams-Bashforth (AB2)

Figure 21.11: The stability diagrams for Adams-Bashforth methods.

For the Crank-Nicolson scheme, we have $\alpha_0 = 1$, $\alpha_1 = -1$, $\beta_0 = 1/2$, and $\beta_1 = 1/2$. Thus, the polynomial is

$$q(x) = \left(1 - \frac{1}{2}(\lambda\Delta t)\right) x + \left(-1 - \frac{1}{2}(\lambda\Delta t)\right) .$$

The root of the polynomial is

$$r = \frac{2 + (\lambda\Delta t)}{2 - (\lambda\Delta t)} .$$

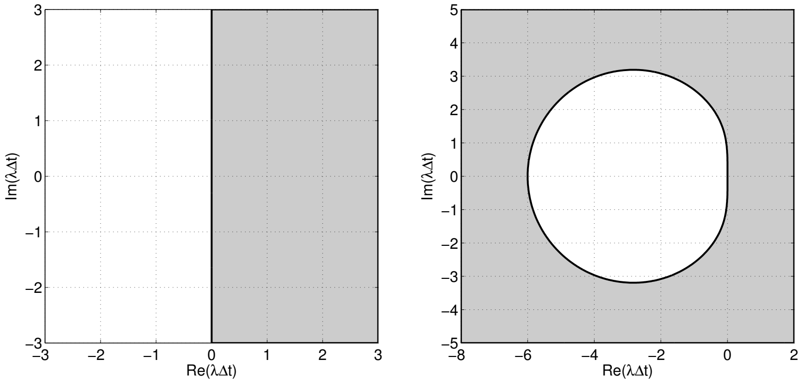
To solve for the stability boundary, let us set $|r| = 1 = |e^{i\theta}|$ and solve for $(\lambda\Delta t)$, i.e.

$$\frac{2 + (\lambda\Delta t)}{2 - (\lambda\Delta t)} = e^{i\theta} \quad \Rightarrow \quad (\lambda\Delta t) = \frac{2(e^{i\theta} - 1)}{e^{i\theta} + 1} = \frac{i2\sin(\theta)}{1 + \cos(\theta)} .$$

Thus, as θ varies from 0 to $\pi/2$, $\lambda\Delta t$ varies from 0 to $i\infty$ along the imaginary axis. Similarly, as θ varies from 0 to $-\pi/2$, $\lambda\Delta t$ varies from 0 to $-i\infty$ along the imaginary axis. Thus, the stability boundary is the imaginary axis. The absolute stability region is the entire left-hand (complex) plane.

The stability diagrams for the 1- and 2-step Adams-Moulton methods are shown in Figure 21.11. The Crank-Nicolson scheme shows the ideal stability diagram; it is stable for all stable ODEs ($\lambda \leq 0$) and unstable for all unstable ODEs ($\lambda > 0$) regardless of the time step selection. (Furthermore, for neutrally stable ODEs, $\lambda = 0$, Crank-Nicolson is neutrally stable — γ, the amplification factor, is unity.) The selection of time step is dictated by the accuracy requirement rather than stability concerns.[1] Despite being an implicit scheme, AM2 is not stable for all $\lambda\Delta t$ in the left-hand plane; for example, along the real axis, the time step is limited to $-\lambda\Delta t \leq 6$. While the stability region is larger than, for example, the Euler Forward scheme, the stability region of AM2 is rather disappointing considering the additional computational cost associated with each step of an implicit scheme.

[1] However, the Crank-Nicolson method does exhibit undesirable oscillations for $\lambda\Delta t \to -$ (real) ∞, and the lack of any dissipation on the imaginary axis can also sometimes cause difficulties. Nobody's perfect.



(a) Crank-Nicolson (AM1)

(b) 2-step Adams-Moulton (AM2)

Figure 21.12: The stability diagrams for 2-step Adams-Moulton methods.

Backward Differentiation Formulas

The Backward Differentiation Formulas are implicit multistep schemes that are well suited for stiff problems. Unlike the Adams-Bashforth and Adams-Moulton schemes, we restrict ourselves to

$$\beta_k = 0, \quad k = 1, \ldots, K \ .$$

Thus, the Backward Differential Formulas are of the form

$$\tilde{u}^j + \sum_{k=1}^{K} \alpha_k \tilde{u}^{j-k} = \Delta t \, \beta_0 g^j \ .$$

Our task is to find the coefficients α_k, $k = 1, \ldots, K$, and β_0. We first construct a K^{th}-degree interpolating polynomial using $\tilde{u}^{j-k}$, $k = 0, \ldots, K$, to approximate $u(t)$, i.e.

$$u(t) \approx \sum_{k=0}^{K} \phi_k(t) \tilde{u}^{j-k} \ ,$$

where $\phi_k(t)$, $k = 0, \ldots, K$, are the Lagrange interpolation polynomials defined at the points t^{j-k}, $k = 0, \ldots, K$; i.e., the same polynomials used to develop the Adams-Moulton schemes. Differentiating the function and evaluating it at $t = t^j$, we obtain

$$\left.\frac{du}{dt}\right|_{t^j} \approx \sum_{k=0}^{K} \left.\frac{d\phi_k}{dt}\right|_{t^j} \tilde{u}^{j-k} \ .$$

Again, we apply the change of variable of the form $t = t^j - \Delta t \hat{\tau}$, so that

$$\left.\frac{du}{dt}\right|_{t^j} \approx \sum_{k=0}^{K} \left.\frac{d\hat{\phi}_k}{d\hat{\tau}}\right|_{0} \left.\frac{d\hat{\tau}}{dt}\right|_{t^j} \tilde{u}^{j-k} = -\frac{1}{\Delta t} \sum_{k=0}^{K} \left.\frac{d\hat{\phi}_k}{d\hat{\tau}}\right|_{0} \tilde{u}^{j-k} \ .$$

Recalling $g^j = g(u(t^j), t^j) = du/dt|_{t^j}$, we set

$$\tilde{u}^j + \sum_{k=1}^{K} \alpha_k \tilde{u}^{j-k} \approx \Delta t \beta_0 \left(-\frac{1}{\Delta t} \sum_{k=0}^{K} \left.\frac{d\hat{\phi}_k}{d\hat{\tau}}\right|_{0} \tilde{u}^{j-k} \right) = -\beta_0 \sum_{k=0}^{K} \left.\frac{d\hat{\phi}_k}{d\hat{\tau}}\right|_{0} \tilde{u}^{j-k} \ .$$

Matching the coefficients for $\tilde{u}^{j-k}$, $k = 0, \ldots, K$, we obtain

$$1 = -\beta_0 \left.\frac{d\hat{\phi}_k}{d\hat{\tau}}\right|_0$$

$$\alpha_k = -\beta_0 \left.\frac{d\hat{\phi}_k}{d\hat{\tau}}\right|_0 , \quad k = 1, \ldots, K \ .$$

Let us develop a few Backward Differentiation Formulas.

Example 21.1.11 1-step Backward Differentiation Formula (Euler Backward)

The 1-step Backward Differentiation Formula requires specification of β_0 and α_1. As in the 1-step Adams-Moulton scheme, the Lagrange polynomials for this case are

$$\hat{\phi}_0(\hat{\tau}) = -\tau + 1 \quad \text{and} \quad \hat{\phi}_1(\hat{\tau}) = \tau \ .$$

Differentiating and evaluating at $\hat{\tau} = 0$

$$\beta_0 = -\left(\left.\frac{d\hat{\phi}_0}{d\hat{\tau}}\right|_0 \right)^{-1} = -(-1)^{-1} = 1 \ ,$$

$$\alpha_1 = -\beta_0 \left.\frac{d\hat{\phi}_1}{d\hat{\tau}}\right|_0 = -1 \ .$$

The resulting scheme is

$$\tilde{u}^j - \tilde{u}^{j-1} = \Delta t g^j \ ,$$

which is the Euler Backward scheme. Again.

·

Example 21.1.12 2-step Backward Differentiation Formula

The 2-step Backward Differentiation Formula requires specification of β_0, α_1, and α_2. The Lagrange polynomials for this case are

$$\hat{\phi}_0(\hat{\tau}) = \frac{1}{2}(\hat{\tau}^2 - 3\hat{\tau} + 2) \ ,$$

$$\hat{\phi}_1(\hat{\tau}) = -\hat{\tau}^2 + 2\hat{\tau} \ ,$$

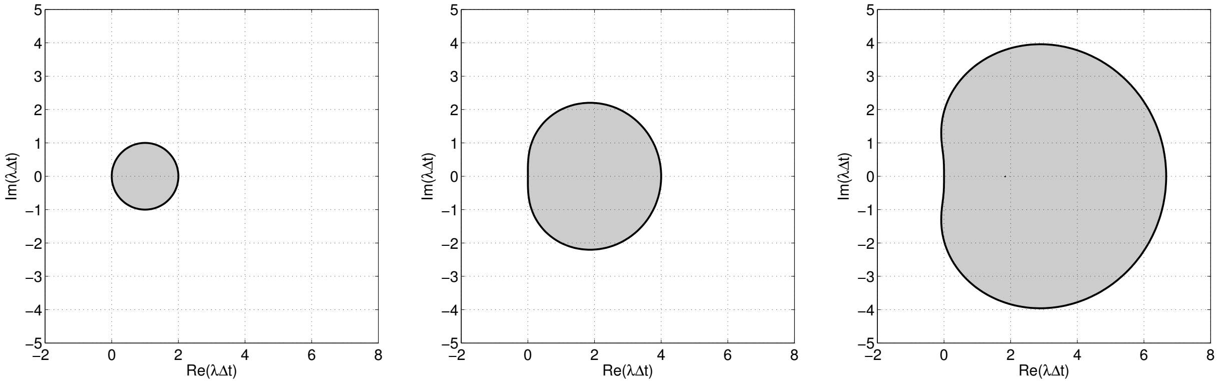
$$\hat{\phi}_2(\hat{\tau}) = \frac{1}{2}\left(\hat{\tau}^2 - \hat{\tau}\right) .$$

Differentiation yields

$$\beta_0 = -\left(\left.\frac{d\hat{\phi}_0}{d\hat{\tau}}\right|_0 \right)^{-1} = \frac{2}{3} \ ,$$

$$\alpha_1 = -\beta_0 \left.\frac{d\hat{\phi}_1}{d\hat{\tau}}\right|_0 = -\frac{2}{3} \cdot 2 = -\frac{4}{3} \ ,$$

$$\alpha_2 = -\beta_0 \left.\frac{d\hat{\phi}_2}{d\hat{\tau}}\right|_0 = -\frac{2}{3} \cdot -\frac{1}{2} = \frac{1}{3} \ .$$



(a) BDF1 (Euler Backward)

(b) BDF2

(c) BDF3

Figure 21.13: The absolute stability diagrams for Backward Differentiation Formulas.

The resulting scheme is

$$\tilde{u}^j - \frac{4}{3}\tilde{u}^{j-1} + \frac{1}{3}\tilde{u}^{j-2} = \frac{2}{3}\Delta t g^j \ .$$

The 2-step Backward Differentiation Formula (BDF2) is unconditionally stable and is second-order accurate.

Example 21.1.13 3-step Backward Differentiation Formula

Following the same procedure, we can develop the 3-step Backward Differentiation Formula (BDF3). The scheme is given by

$$\tilde{u}^j - \frac{18}{11}\tilde{u}^{j-1} + \frac{9}{11}\tilde{u}^{j-2} - \frac{2}{11}\tilde{u}^{j-3} = \frac{6}{11}\Delta t g^j \ .$$

The scheme is unconditionally stable and is third-order accurate.

The stability diagrams for the 1-, 2-, and 3-step Backward Differentiation Formulas are shown in Figure 21.13. The BDF1 and BDF2 schemes are A-stable (i.e., the stable region includes the entire left-hand plane). Unfortunately, BDF3 is not A-stable; in fact the region of instability in the left-hand plane increases for the higher-order BDFs. However, for stiff engineering systems whose eigenvalues are clustered along the real axis, the BDF methods are attractive choices.

21.1.9 Multistage Schemes: Runge-Kutta

Another family of important and powerful integration schemes are multistage schemes, the most famous of which are the Runge-Kutta schemes. While a detailed analysis of the Runge-Kutta schemes is quite involved, we briefly introduce the methods due to their prevalence in the scientific and engineering context.

Unlike multistep schemes, multistage schemes only require the solution at the previous time step $\tilde{u}^{j-1}$ to approximate the new state $\tilde{u}^j$ at time t^j. To develop an update formula, we first observe that

$$u(t^j) = \tilde{u}(t^{j-1}) + \int_{t^{j-1}}^{t^j} \frac{du}{dt}(\tau)d\tau = \tilde{u}(t^{j-1}) + \int_{t^{j-1}}^{t^j} g(u(\tau),\tau)d\tau \ .$$

Clearly, we cannot use the formula directly to approximate $u(t^j)$ because we do not know $g(u(\tau), \tau)$, $\tau \in]t^{j-1}, t^j[$. To derive the Adams schemes, we replaced the unknown function g with its polynomial approximation based on g evaluated at K previous time steps. In the case of Runge-Kutta, we directly apply numerical quadrature to the integral to obtain

$$u(t^j) \approx u(t^{j-1}) + \Delta t \sum_{k=1}^{K} b_k \, g\left(u(t^{j-1} + c_k \Delta t), t^{j-1} + c_k \Delta t\right) \ ,$$

where the b_k are the quadrature weights and the $t^j + c_k \Delta t$ are the quadrature points. We need to make further approximations to define a scheme, because we do not know the values of u at the K stages, $u(t^j + c_k \Delta t)$, $k = 1, \ldots, K$. Our approach is to replace the K stage values $u(t^{j-1} + c_k \Delta t)$ by approximations v_k and then to form the K stage derivatives as

$$G_k = g\left(v_k, t^{j-1} + c_k \Delta t\right) \ .$$

It remains to specify the approximation scheme.

For an explicit Runge-Kutta scheme, we construct the k^{th}-stage approximation as a linear combination of the previous stage derivatives and $\tilde{u}^{j-1}$, i.e.

$$v_k = \tilde{u}^{j-1} + \Delta t \left(A_{k1} G_1 + A_{k2} G_2 + \cdots + A_{k,k-1} G_{k-1}\right) .$$

Because this k^{th}-stage estimate only depends on the previous stage derivatives, we can compute the stage values in sequence,

$$\begin{aligned}
v_1 &= \tilde{u}^{j-1} & &(\Rightarrow G_1) \ , \\
v_2 &= \tilde{u}^{j-1} + \Delta t A_{21} G_1 & &(\Rightarrow G_2) \ , \\
v_3 &= \tilde{u}^{j-1} + \Delta t A_{31} G_1 + \Delta t A_{32} G_2 & &(\Rightarrow G_3) \ , \\
&\vdots \\
v_K &= \tilde{u}^{j-1} + \Delta t \textstyle\sum_{k=1}^{K-1} A_{Kk} G_k & &(\Rightarrow G_K) \ .
\end{aligned}$$

Once the stage values are available, we estimate the integral by

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t \sum_{k=1}^{K} b_k \, G_k \ ,$$

and proceed to the next time step.

Note that a Runge-Kutta scheme is uniquely defined by the choice of the vector b for quadrature weight, the vector c for quadrature points, and the matrix A for the stage reconstruction. The coefficients are often tabulated in a Butcher table, which is a collection of the coefficients of the form

$$\begin{array}{c|c} c & A \\ \hline & b^{\mathrm{T}} \end{array} \ .$$

For explicit Runge-Kutta methods, we require $A_{ij} = 0$, $i \leq j$. Let us now introduce two popular explicit Runge-Kutta schemes.

Example 21.1.14 Two-stage Runge-Kutta
A popular two-stage Runge-Kutta method (RK2) has the Butcher table

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} .$$

This results in the following update formula

$$\begin{aligned} v_1 &= \tilde{u}^{j-1}, & G_1 &= g(v_1, t^{j-1}) \ , \\ v_2 &= \tilde{u}^{j-1} + \tfrac{1}{2}\Delta t G_1, & G_2 &= g\left(v_2, t^{j-1} + \frac{1}{2}\Delta t\right) \ , \\ \tilde{u}^j &= \tilde{u}^j + \Delta t G_2 \ . \end{aligned}$$

The two-stage Runge-Kutta scheme is conditionally stable and is second-order accurate. We might view this scheme as a kind of midpoint rule.

Example 21.1.15 Four-stage Runge-Kutta
A popular four-stage Runge-Kutta method (RK4) — and perhaps the most popular of all Runge-Kutta methods — has the Butcher table of the form

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} .$$

This results in the following update formula

$$\begin{aligned} v_1 &= \tilde{u}^{j-1}, & G_1 &= g(v_1, t^{j-1}) \ , \\ v_2 &= \tilde{u}^{j-1} + \tfrac{1}{2}\Delta t G_1, & G_2 &= g\left(v_2, t^{j-1} + \frac{1}{2}\Delta t\right), \\ v_3 &= \tilde{u}^{j-1} + \tfrac{1}{2}\Delta t G_2, & G_3 &= g\left(v_3, t^{j-1} + \frac{1}{2}\Delta t\right), \\ v_4 &= \tilde{u}^{j-1} + \Delta t G_3, & G_4 &= g\left(v_4, t^{j-1} + \Delta t\right), \\ \tilde{u}^j &= \tilde{u}^{j-1} + \Delta t \left(\frac{1}{6}G_1 + \frac{1}{3}G_2 + \frac{1}{3}G_3 + \frac{1}{6}G_4\right). \end{aligned}$$

The four-stage Runge-Kutta scheme is conditionally stable and is fourth-order accurate.

The accuracy analysis of the Runge-Kutta schemes is quite involved and is omitted here. There are various choices of coefficients that achieve p^{th}-order accuracy using p stages for $p \le 4$. It is also worth noting that even though we can achieve fourth-order accuracy using a four-stage Runge-Kutta method, six stages are necessary to achieve fifth-order accuracy.

Explicit Runge-Kutta methods required that a stage value is a linear combination of the previous stage derivatives. In other words, the A matrix is lower triangular with zeros on the diagonal. This made the calculation of the state values straightforward, as we could compute the stage values in sequence. If we remove this restriction, we arrive at family of *implicit Runge-Kutta methods* (IRK). The stage value updates for implicit Runge-Kutta schemes are fully coupled, i.e.

$$v_k = \tilde{u}^{j-1} + \Delta t \sum_{i=1}^{K} A_{ki} G_i, \quad k = 1, \dots, K \ .$$

In other words, the matrix A is full in general. Like other implicit methods, implicit Runge-Kutta schemes tend to be more stable than their explicit counterparts (although also more expensive per time step). Moreover, for all K, there is a unique IRK method that achieves $2K$ order of accuracy. Let us introduce one such scheme.

Example 21.1.16 Two-stage Gauss-Legendre Implicit Runge-Kutta
The two-stage Gauss-Legendre Runge-Kutta method[2] (GL-IRK2) is described by the Butcher table

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \ .$$

To compute the update we must first solve a system of equations to obtain the stage values v_1 and v_2

$$v_1 = \tilde{u}^{j-1} + A_{11}\Delta t G_1 + A_{12}\Delta G_2 \ ,$$

$$v_2 = \tilde{u}^{j-1} + A_{21}\Delta t G_1 + A_{12}\Delta G_2 \ ,$$

or

$$v_1 = \tilde{u}^{j-1} + A_{11}\Delta t g(v_1, t^{j-1} + c_1\Delta t) + A_{12}\Delta t g(v_2, t^{j-1} + c_2\Delta t) \ ,$$

$$v_2 = \tilde{u}^{j-1} + A_{21}\Delta t g(v_1, t^{j-1} + c_1\Delta t) + A_{22}\Delta t g(v_2, t^{j-1} + c_2\Delta t) \ ,$$

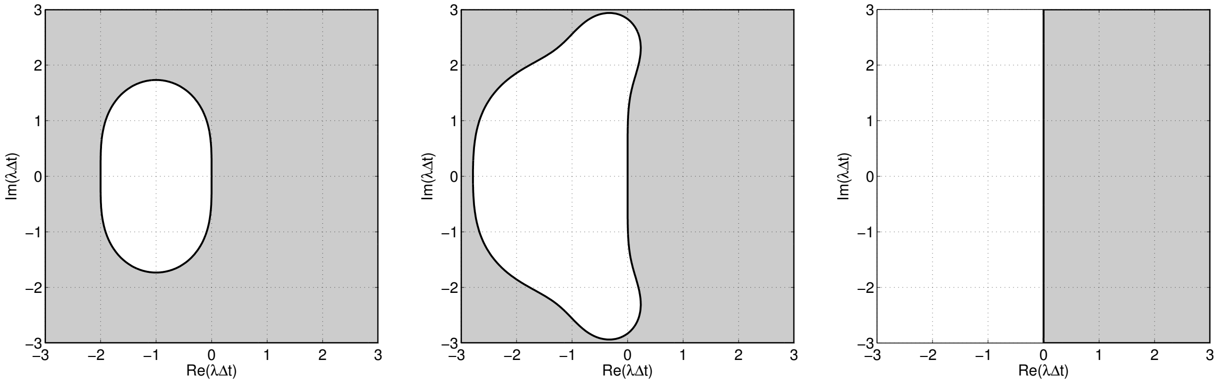
where the coefficients A and c are provided by the Butcher table. Once the stage values are computed, we compute $\tilde{u}^j$ according to

$$\tilde{u}^j = \tilde{u}^{j-1} + \Delta t \quad b_1\, g(v_1, t^{j-1} + c_1\Delta t) + b_2\, g(v_2, t^{j-1} + c_2\Delta t) \quad ,$$

where the coefficients b are given by the Butcher table.

The two-stage Gauss-Legendre Runge-Kutta scheme is A-stable and is fourth-order accurate. While the method is computationally expensive and difficult to implement, the A-stability and fourth-order accuracy are attractive features for certain applications.

[2] The naming is due to the use of the Gauss quadrature points, which are the roots of Legendre polynomials on the unit interval.



(a) RK2 (b) RK4 (c) Gauss-Legendre IRK2

Figure 21.14: The absolute stability diagrams for the Runge-Kutta family of schemes.

There is a family of implicit Runge-Kutta methods called *diagonally implicit Runge-Kutta* (DIRK). These methods have an A matrix that is lower triangular with the same coefficients in each diagonal element. This family of methods inherits the stability advantage of IRK schemes while being computationally more efficient than other IRK schemes for nonlinear systems, as we can incrementally update the stages.

The stability diagrams for the three Runge-Kutta schemes presented are shown in Figure 21.14. The two explicit Runge-Kutta methods, RK2 and RK4, are not A-stable. The time step along the real axis is limited to $-\lambda\Delta t \leq 2$ for RK2 and $-\lambda\Delta t \lesssim 2.8$ for RK4. However, the stability region for the explicit Runge-Kutta schemes are considerably larger than the Adams-Bashforth family of explicit schemes. While the explicit Runge-Kutta methods are not suited for very stiff systems, they can be used for moderately stiff systems. The implicit method, GL-IRK2, is A-stable; it also correctly exhibits growing behavior for unstable systems.

Figure 21.15 shows the error behavior of the Runge-Kutta schemes applied to $du/dt = -4u$. The higher accuracy of the Runge-Kutta schemes compared to the Euler Forward scheme is evident from the solution. The error convergence plot confirms the theoretical convergence rates for these methods.

21.2 Scalar Second-Order Linear ODEs

21.2.1 Model Problem

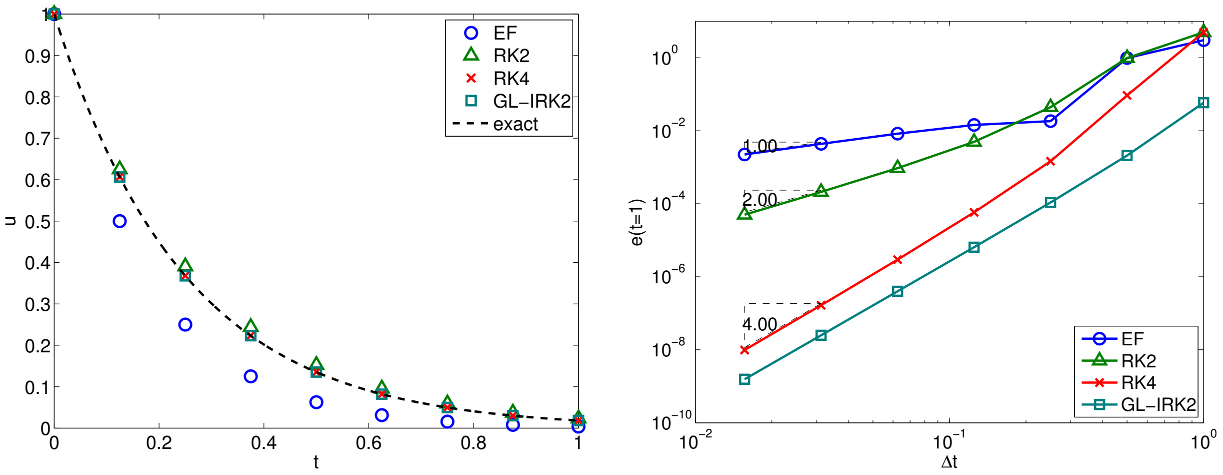
Let us consider a canonical second-order ODE,

$$m\frac{d^2u}{dt^2} + c\frac{du}{dt} + ku = f(t), \quad 0 < t < t_f \ ,$$

$$u(0) = u_0 \ ,$$

$$\frac{du}{dt}(0) = v_0 \ .$$

The ODE is second order, because the highest derivative that appears in the equation is the second derivative. Because the equation is second order, we now require *two* initial conditions: one for



(a) solution ($\Delta t = 1/8$)

(b) error

Figure 21.15: The error convergence behavior for the Runge-Kutta family of schemes applied to $du/dt = -4u$. Here $e(t = 1) = |u(t^j) - \tilde{u}^j|$ for $t^j = j\Delta t = 1$.

displacement, and one for velocity. It is a linear ODE because the equation is linear with respect to u and its derivatives.

A typical spring-mass-damper system is governed by this second-order ODE, where u is the displacement, m is the mass, c is the damping constant, k is the spring constant, and f is the external forcing. This system is of course a damped oscillator, as we now illustrate through the classical solutions.

21.2.2 Analytical Solution

Homogeneous Equation: Undamped

Let us consider the undamped homogeneous case, with $c = 0$ and $f = 0$,

$$m\frac{d^2u}{dt^2} + ku = 0, \quad 0 < t < t_f \ ,$$

$$u(0) = u_0 \ ,$$

$$\frac{du}{dt}(0) = v_0 \ .$$

To solve the ODE, we assume solutions of the form $e^{\lambda t}$, which yields

$$(m\lambda^2 + k)\, e^{\lambda t} = 0 \ .$$

This implies that $m\lambda^2 + k = 0$, or that λ must be a root of the characteristic polynomial

$$p(\lambda) = m\lambda^2 + k = 0 \quad \Rightarrow \quad \lambda_{1,2} = \pm i\sqrt{\frac{k}{m}} \ .$$

Let us define the *natural frequency*, $\omega_n \equiv \sqrt{k/m}$. The roots of the characteristic polynomials are then $\lambda_{1,2} = \pm i\omega_n$. The solution to the ODE is thus of the form

$$u(t) = \alpha e^{i\omega_n t} + \beta e^{-i\omega_n t} \ .$$

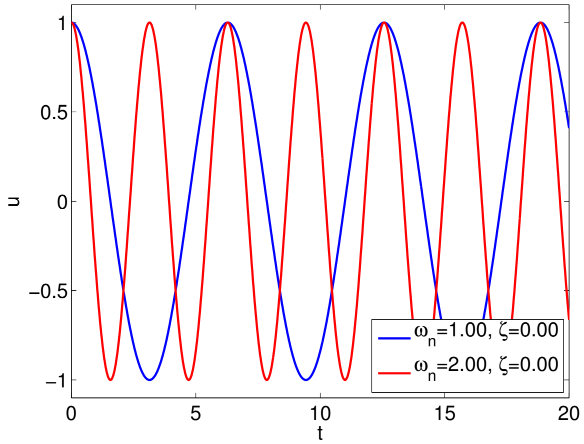


Figure 21.16: Response of undamped spring-mass systems.

Rearranging the equation,

$$u(t) = \alpha e^{i\omega_n t} + \beta e^{-i\omega_n t} = \frac{\alpha+\beta}{2}(e^{i\omega_n t} + e^{-i\omega_n t}) + \frac{\alpha-\beta}{2}(e^{i\omega_n t} - e^{-i\omega_n t})$$

$$= (\alpha+\beta)\ \cos(\omega_n t) + i(\alpha-\beta)\ \sin(\omega_n t)\ .$$

Without loss of generality, let us redefine the coefficients by $c_1 = \alpha + \beta$ and $c_2 = i(\alpha - \beta)$. The general form of the solution is thus

$$u(t) = c_1\ \cos(\omega_n t) + c_2\ \sin(\omega_n t)\ .$$

The coefficients c_1 and c_2 are specified by the initial condition. In particular,

$$\begin{aligned} u(t=0) = c_1 = u_0 \qquad &\Rightarrow \qquad c_1 = u_0\ , \\ \frac{du}{dt}(t=0) = c_2\omega_n = v_0 \qquad &\Rightarrow \qquad c_2 = \frac{v_0}{\omega_n}\ . \end{aligned}$$

Thus, the solution to the undamped homogeneous equation is

$$u(t) = u_0 \cos(\omega_n t) + \frac{v_0}{\omega_n} \sin(\omega_n t)\ ,$$

which represents a (non-decaying) sinusoid.

Example 21.2.1 Undamped spring-mass system

Let us consider two spring-mass systems with the natural frequencies $\omega_n = 1.0$ and 2.0. The responses of the systems to initial displacement of $u(t = 0) = 1.0$ are shown in Figure 21.16. As the systems are undamped, the amplitudes of the oscillations do not decay with time.

·

Homogeneous Equation: Underdamped

Let us now consider the homogeneous case ($f = 0$) but with finite (but weak) damping

$$m\frac{d^2u}{dt^2} + c\frac{du}{dt} + ku = 0, \quad 0 < t < t_f \ ,$$

$$u(0) = u_0 \ ,$$

$$\frac{du}{dt}(0) = v_0 \ .$$

To solve the ODE, we again assume behavior of the form $u = e^{\lambda t}$. Now the roots of the characteristic polynomial are given by

$$p(\lambda) = m\lambda^2 + c\lambda + k = 0 \quad \Rightarrow \quad \lambda_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} \ .$$

Let us rewrite the roots as

$$\lambda_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}} = -\sqrt{\frac{k}{m}}\frac{c}{2\sqrt{mk}} \pm \sqrt{\frac{k}{m}}\sqrt{\frac{c^2}{4mk} - 1} \ .$$

For convenience, let us define the *damping ratio* as

$$\zeta = \frac{c}{2\sqrt{mk}} = \frac{c}{2m\omega_n} \ .$$

Together with the definition of natural frequency, $\omega_n = \sqrt{k/m}$, we can simplify the roots to

$$\lambda_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \ .$$

The underdamped case is characterized by the condition

$$\zeta^2 - 1 < 0 \ ,$$

i.e., $\zeta < 1$.

In this case, the roots can be conveniently expressed as

$$\lambda_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1 - \zeta^2} = -\zeta\omega_n \pm i\omega_d \ ,$$

where $\omega_d \equiv \omega_n\sqrt{1-\zeta^2}$ is the damped frequency. The solution to the underdamped homogeneous system is

$$u(t) = \alpha e^{-\zeta\omega_n t + i\omega_d t} + \beta e^{-\zeta\omega_n t - i\omega_d t} \ .$$

Using a similar technique as that used for the undamped case, we can simplify the expression to

$$u(t) = e^{-\zeta\omega_n t} \ \ c_1 \ \cos(\omega_d t) + c_2 \ \sin(\omega_d t) \ \ .$$

Substitution of the initial condition yields

$$u(t) = e^{-\zeta\omega_n t}\left(u_0 \ \cos(\omega_d t) + \frac{v_0 + \zeta\omega_n u_0}{\omega_d} \ \sin(\omega_d t)\right) .$$

Thus, the solution is sinusoidal with exponentially decaying amplitude. The decay rate is set by the damping ratio, ζ. If $\zeta \ll 1$, then the oscillation decays slowly — over many periods.

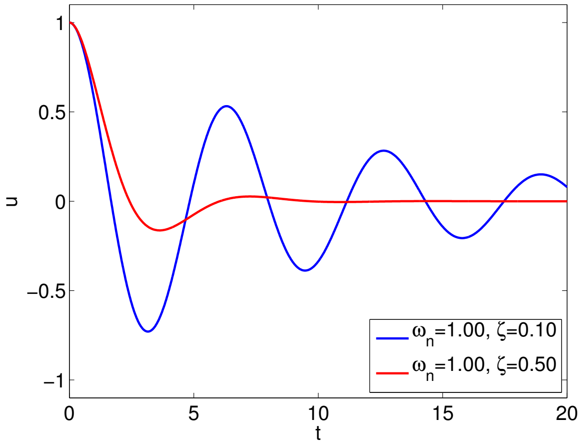


Figure 21.17: Response of underdamped spring-mass-damper systems.

Example 21.2.2 Underdamped spring-mass-damper system
Let us consider two underdamped spring-mass-damper systems with

$$\begin{aligned}&\text{System 1:} \quad \omega_n = 1.0 \quad \text{and} \quad \zeta = 0.1\\&\text{System 2:} \quad \omega_n = 1.0 \quad \text{and} \quad \zeta = 0.5\,.\end{aligned}$$

The responses of the systems to initial displacement of $u(t = 0) = 1.0$ are shown in Figure 21.17. Unlike the undamped systems considered in Example 21.2.1, the amplitude of the oscillations decays with time; the oscillation of System 2 with a higher damping coefficient decays quicker than that of System 1.

·

Homogeneous Equation: Overdamped

In the underdamped case, we assumed $\zeta < 1$. If $\zeta > 1$, then we have an *overdamped* system. In this case, we write the roots as

$$\lambda_{1,2} = -\omega_n \left(\zeta \pm \sqrt{\zeta^2 - 1}\right) ,$$

both of which are real. The solution is then given by

$$u(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} .$$

The substitution of the initial conditions yields

$$c_1 = \frac{\lambda_2 u_0 - v_0}{\lambda_2 - \lambda_1} \quad \text{and} \quad c_2 = \frac{-\lambda_2 u_0 + v_0}{\lambda_2 - \lambda_1} .$$

The solution is a linear combination of two exponentials that decay with time constants of $1/|\lambda_1|$ and $1/|\lambda_2|$, respectively. Because $|\lambda_1| > |\lambda_2|$, $|\lambda_2|$ dictates the long time decay behavior of the system. For $\zeta \to \infty$, λ_2 behaves as $-\omega_n/(2\zeta) = -k/c$.

Example 21.2.3 Overdamped spring-mass-damper system
Let us consider two overdamped spring-mass-damper systems with

$$\begin{aligned}&\text{System 1:} \quad \omega_n = 1.0 \quad \text{and} \quad \zeta = 1.0\\&\text{System 2:} \quad \omega_n = 1.0 \quad \text{and} \quad \zeta = 5.0\,.\end{aligned}$$

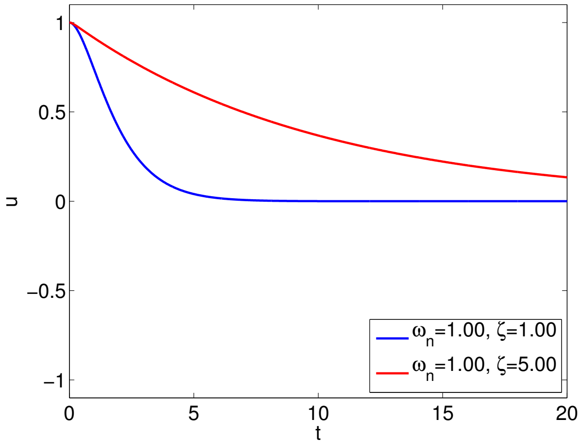


Figure 21.18: Response of overdamped spring-mass-damper systems.

The responses of the systems to initial displacement of $u(t=0)=1.0$ are shown in Figure 21.17. As the systems are overdamped, they exhibit non-oscillatory behaviors. Note that the oscillation of System 2 with a higher damping coefficient decays more slowly than that of System 1. This is in contrast to the underdamped cases considered in Example 21.2.2, in which the oscillation of the system with a higher damping coefficient decays more quickly.

·

Sinusoidal Forcing

Let us consider a sinusoidal forcing of the second-order system. In particular, we consider a system of the form

$$m\frac{d^2u}{dt^2}+c\frac{du}{dt}+ku=A\cos(\omega t)\ .$$

In terms of the natural frequency and the damping ratio previously defined, we can rewrite the system as

$$\frac{d^2u}{dt^2}+2\zeta\omega_n\frac{du}{dt}+\omega_n^2u=\frac{A}{m}\ \cos(\omega t)\ .$$

A particular solution is of the form

$$u_p(t)=\alpha\cos(\omega t)+\beta\sin(\omega t)\ .$$

Substituting the assumed form of particular solution into the governing equation, we obtain

$$\begin{aligned}0&=\frac{d^2u_p}{dt^2}+2\zeta\omega_n\frac{du_p}{dt}+\omega_n^2u_p-\frac{A}{m}\ \cos(\omega t)\\&=-\alpha\omega^2\cos(\omega t)-\beta\omega^2\sin(\omega t)+2\zeta\omega_n(-\alpha\omega\sin(\omega t)+\beta\omega\cos(\omega t))\\&\quad+\omega_n^2(\alpha\cos(\omega t)+\beta\sin(\omega t))-A\cos(\omega t)\ .\end{aligned}$$

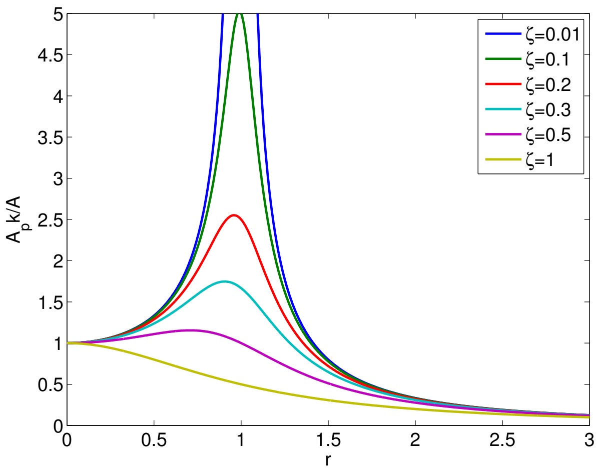


Figure 21.19: The variation in the amplification factor for the sinusoidally forced system.

We next match terms in sin and cos to obtain

$$\alpha(\omega_n^2 - \omega^2) + \beta(2\zeta\omega\omega_n) = \frac{A}{m} ,$$

$$\beta(\omega_n^2 - \omega^2) - \alpha(2\zeta\omega\omega_n) = 0 ,$$

and solve for the coefficients,

$$\alpha = \frac{(\omega_n^2 - \omega^2)}{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega\omega_n)^2}\frac{A}{m} = \frac{1 - r^2}{(1 - r^2)^2 + (2\zeta r)^2}\frac{A}{m\omega_n^2} = \frac{1 - r^2}{(1 - r^2)^2 + (2\zeta r)^2}\frac{A}{k} ,$$

$$\beta = \frac{(2\zeta\omega\omega_n)}{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega\omega_n)^2}\frac{A}{m} = \frac{2\zeta r}{(1 - r^2)^2 + (2\zeta r)^2}\frac{A}{m\omega_n^2} = \frac{2\zeta r}{(1 - r^2)^2 + (2\zeta r)^2}\frac{A}{k} ,$$

where $r \equiv \omega/\omega_n$ is the ratio of the forced to natural frequency.

Using a trigonometric identity, we may compute the amplitude of the particular solution as

$$A_p = \sqrt{\alpha^2 + \beta^2} = \frac{\sqrt{(1 - r^2)^2 + (2\zeta r)^2}}{(1 - r^2)^2 + (2\zeta r)^2}\frac{A}{k} = \frac{1}{\sqrt{(1 - r^2)^2 + (2\zeta r)^2}}\frac{A}{k} .$$

Note that the magnitude of the amplification varies with the frequency ratio, r, and the damping ratio, ζ. This variation in the amplification factor is plotted in Figure 21.19. For a given ζ, the amplification factor is maximized at $r = 1$ (i.e., $\omega_n = \omega$), and the peak amplification factor is $1/(2\zeta)$. This increase in the magnitude of oscillation near the natural frequency of the system is known as *resonance*. The natural frequency is clearly crucial in understanding the forced response of the system, in particular for lightly damped systems.[3]

21.3 System of Two First-Order Linear ODEs

It is possible to directly numerically tackle the second-order system of Section 21.2 for example using Newmark integration schemes. However, we shall focus on a state-space approach which is much more general and in fact is the basis for numerical solution of systems of ODEs of virtually any kind.

[3] Note that for $\zeta = 0$ (which in fact is not realizable physically in any event), the amplitude is only infinite as $t \to \infty$; in particular, in resonant conditions, the amplitude will grow linearly in time.

21.3.1 State Space Representation of Scalar Second-Order ODEs

In this section, we develop a state space representation of the canonical second-order ODE. Recall that the ODE of interest is of the form

$$\frac{d^2u}{dt^2} + 2\zeta\omega_n \frac{du}{dt} + \omega_n^2 u = \frac{1}{m}\, f(t), \quad 0 < t < t_f \ ,$$

$$u(0) = u_0 \ ,$$

$$\frac{du}{dt}(0) = v_0 \ .$$

Because this is a second-order equation, we need two variables to fully describe the state of the system. Let us choose these state variables to be

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

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

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

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

$$\frac{dw_2}{dt} = \frac{d}{dt}\frac{du}{dt} = \frac{d^2u}{dt^2} - 2\zeta\omega_n \frac{du}{dt} = -\omega_n^2 u + \frac{1}{m}\, f = -2\zeta\omega_n w_2 - \omega_n^2 w_1 + \frac{1}{m}\, f \ .$$

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

$$\frac{d}{dt}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} w_2 \\ -\omega_n^2 w_1 - 2\zeta\omega_n w_2 + \frac{1}{m}\, f \end{pmatrix} .$$

This equation can be written in the matrix form

$$\frac{d}{dt}\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{pmatrix}}_{A} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m}\, f \end{pmatrix} \tag{21.2}$$

with the initial condition

$$w_1(0) = u_0 \quad \text{and} \quad w_2(0) = v_0 \ .$$

If we define $w = (w_1 \ \ w_2)^{\mathrm{T}}$ and $F = (0 \ \ \frac{1}{m} f)^{\mathrm{T}}$, then

$$\frac{dw}{dt} = Aw + F, \quad w(t = 0) = w_0 = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \ , \tag{21.3}$$

succinctly summarizes the “state-space” representation of our ODE.

Solution by Modal Expansion

To solve this equation, we first find the eigenvalues of A. We recall that the eigenvalues are the roots of the characteristic equation $p(\lambda; A) = \det(\lambda I - A)$, where det refers to the determinant. (In actual practice for large systems the eigenvalues are not computed from the characteristic equation. In our 2×2 case we obtain

$$p(\lambda; A) = \det(\lambda I - A) = \det \begin{pmatrix} \lambda & -1 \\ \omega_n^2 & \lambda + 2\zeta\omega_n \end{pmatrix} = \lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \ .$$

The eigenvalues, the roots of characteristic equation, are thus

$$\lambda_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \ .$$

We shall henceforth assume that the system is underdamped (i.e., $\zeta < 1$), in which case it is more convenient to express the eigenvalues as

$$\lambda_{1,2} = -\zeta\omega_n \pm i\omega_n\sqrt{1 - \zeta^2} \ .$$

Note since the eigenvalue has non-zero imaginary part the solution will be oscillatory and since the real part is negative (left-hand of the complex plane) the solution is stable. We now consider the eigenvectors.

Towards that end, we first generalize our earlier discussion of vectors of real-valued components to the case of vectors of complex-valued components. To wit, if we are given two vectors $v \in \mathbb{C}^{m\times 1}$, $w \in \mathbb{C}^{m\times 1}$ — v and w are each column vectors with m complex entries — the inner product is now given by

$$\beta = v^{\mathrm{H}} w = \sum_{j=1}^{m} v_j^* \, w_j \ , \tag{21.4}$$

where β is in general complex, H stands for Hermitian (complex transpose) and replaces T for transpose, and * denotes complex conjugate — so $v_j = \mathrm{Real}(v_j) + i\,\mathrm{Imag}(v_j)$ and $v_j^* = \mathrm{Real}(v_j) - i\,\mathrm{Imag}(v_j)$, for $i = \sqrt{-1}$.

The various concepts built on the inner product change in a similar fashion. For example, two complex-valued vectors v and w are orthogonal if $v^{\mathrm{H}} w = 0$. Most importantly, the norm of complex-valued vector is now given by

$$\|v\| = \sqrt{v^{\mathrm{H}} v} = \left(\sum_{j=1}^{m} v_j^* \, v_j \right)^{1/2} = \left(\sum_{j=1}^{m} |v_j|^2 \right)^{1/2} \ , \tag{21.5}$$

where $|\cdot|$ denotes the complex modulus; $|v_j|^2 = v_j^* \, v_j = (\mathrm{Real}(v_j))^2 + (\mathrm{Imag}(v_j))^2$. Note the definition (21.5) of the norm ensures that $\|v\|$ is a non-negative real number, as we would expect of a length.

To obtain the eigenvectors, we must find a solution to the equation

$$(\lambda I - A)\chi = 0 \tag{21.6}$$

for $\lambda = \lambda_1$ ($\Rightarrow$ eigenvector $\chi^1 \in \mathbb{C}^2$) and $\lambda = \lambda_2$ ($\Rightarrow$ eigenvector $\chi^2 \in \mathbb{C}^2$). The equations (21.6) will have a solution since λ has been chosen to make $(\lambda I - A)$ singular: the columns of $\lambda I - A$ are *not* linearly independent, and hence there exists a (in fact, many) nontrivial linear combination, $\chi \neq 0$, of the columns of $\lambda I - A$ which yields the zero vector.

Proceeding with the first eigenvector, we write $(\lambda_1 I - A)\chi^1 = 0$ as

$$\begin{pmatrix} -\zeta\omega_n + i\omega_n\sqrt{1-\zeta^2} & -1 \\ \omega_n^2 & \zeta\omega_n + i\omega_n\sqrt{1-\zeta^2} \end{pmatrix} \begin{pmatrix} \chi_1^1 \\ \chi_2^1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

to obtain (say, setting $\chi_1^1 = c$),

$$\chi^1 = c \begin{pmatrix} 1 \\ \dfrac{-\omega_n^2}{\zeta\omega_n + i\omega_n\sqrt{1-\zeta^2}} \end{pmatrix} .$$

We now choose c to achieve $\|\chi^1\| = 1$, yielding

$$\chi^1 = \frac{1}{\sqrt{1+\omega_n^2}} \begin{pmatrix} 1 \\ -\zeta\omega_n + i\omega_n\sqrt{1-\zeta^2} \end{pmatrix} .$$

In a similar fashion we obtain from $(\lambda_2 I - A)\chi^2 = 0$ the second eigenvector

$$\chi^2 = \frac{1}{\sqrt{1+\omega_n^2}} \begin{pmatrix} 1 \\ -\zeta\omega_n - i\omega_n\sqrt{1-\zeta^2} \end{pmatrix} ,$$

which satisfies $\|\chi^2\| = 1$.

We now introduce two additional vectors, ψ^1 and ψ^2. The vector ψ^1 is chosen to satisfy $(\psi^1)^{\mathrm{H}}\chi^2 = 0$ and $(\psi^1)^{\mathrm{H}}\chi^1 = 1$, while the vector ψ^2 is chosen to satisfy $(\psi^2)^{\mathrm{H}}\chi^1 = 0$ and $(\psi^2)^{\mathrm{H}}\chi^2 = 1$. We find, after a little algebra,

$$\psi^1 = \frac{\sqrt{1+\omega_n^2}}{2i\omega_n\sqrt{1-\zeta^2}} \begin{pmatrix} -\zeta\omega_n + i\omega_n\sqrt{1-\zeta^2} \\ -1 \end{pmatrix} , \quad \psi^2 = \frac{\sqrt{1+\omega_n^2}}{-2i\omega_n\sqrt{1-\zeta^2}} \begin{pmatrix} -\zeta\omega_n - i\omega_n\sqrt{1-\zeta^2} \\ -1 \end{pmatrix} .$$

These choices may appear mysterious, but in a moment we will see the utility of this "bi-orthogonal" system of vectors. (The steps here in fact correspond to the "diagonalization" of A.)

We now write w as a linear combination of the two eigenvectors, or "modes,"

$$\begin{aligned} w(t) &= z_1(t)\,\chi^1 + z_2(t)\,\chi^2 \\ &= S\,z(t) \end{aligned} \tag{21.7}$$

where

$$S = (\chi^1 \;\; \chi^2)$$

is the 2×2 matrix whose j^{th}-column is given by the j^{th}-eigenvector, χ^j. We next insert (21.7) into (21.3) to obtain

$$\chi^1 \frac{dz_1}{dt} + \chi^2 \frac{dz_2}{dt} = A(\chi^1 z_1 + \chi^2 z_2) + F , \tag{21.8}$$

$$(\chi^1 z_1 + \chi^2 z_2)(t = 0) = w_0 . \tag{21.9}$$

We now take advantage of the ψ vectors.

First we multiply (21.8) by $(\psi^1)^{\mathrm{H}}$ and take advantage of $(\psi^1)^{\mathrm{H}}\,\chi^2 = 0$, $(\psi^1)^{\mathrm{H}}\,\chi^1 = 1$, and $A\chi^j = \lambda_j \chi^j$ to obtain

$$\frac{dz_1}{dt} = \lambda_1\, z_1 + (\psi^1)^{\mathrm{H}}\, F\ ; \tag{21.10}$$

if we similarly multiply (21.9) we obtain

$$z_1(t=0) = (\psi^1)^{\mathrm{H}}\, w_0\ . \tag{21.11}$$

The same procedure but now with $(\psi^2)^{\mathrm{H}}$ rather than $(\psi^1)^{\mathrm{H}}$ gives

$$\frac{dz_2}{dt} \quad = \quad \lambda_2\, z_2 + (\psi^2)^{\mathrm{H}}\, F\ ; \tag{21.12}$$

$$z_2(t=0) \quad = \quad (\psi^2)^{\mathrm{H}}\, w_0\ . \tag{21.13}$$

We thus observe that our modal expansion reduces our coupled 2×2 ODE system into two decoupled ODEs.

The fact that λ_1 and λ_2 are complex means that z_1 and z_2 are also complex, which might appear inconsistent with our original *real* equation (21.3) and *real* solution $w(t)$. However, we note that $\lambda_2 = \lambda_1^*$ and $\psi^2 = (\psi^1)^*$ and thus $z_2 = z_1^*$. It thus follows from (21.7) that, since $\chi^2 = (\chi^1)^*$ as well,

$$w = z_1\chi^1 + z_1^*(\chi^1)^*\ ,$$

and thus

$$w = 2\,\mathrm{Real}(z_1\chi^1)\ .$$

Upon superposition, our solution is indeed real, as desired.

It is possible to use this modal decomposition to construct numerical procedures. However, our interest here in the modal decomposition is as a way to understand how to choose an ODE scheme for a system of two (later n) ODEs, and, for the chosen scheme, how to choose Δt for stability.

21.3.2 Numerical Approximation of a System of Two ODEs

Crank-Nicolson

The application of the Crank-Nicolson scheme to our system (21.3) is identical to the application of the Crank-Nicolson scheme to a scalar ODE. In particular, we directly take the scheme of example 21.1.8 and replace $\tilde{u}^j \in \mathbb{R}$ with $\tilde{w}^j \in \mathbb{R}^2$ and g with $A\tilde{w}^j + F^j$ to obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \frac{\Delta t}{2}\,(A\tilde{w}^j + A\tilde{w}^{j-1}) + \frac{\Delta t}{2}\,(F^j + F^{j-1})\ . \tag{21.14}$$

(Note if our force f is constant in time then $F^j = F$.) In general if follows from consistency arguments that we will obtain the same order of convergence as for the scalar problem — *if* (21.14) is stable. The difficult issue for systems is *stability*: Will a particular scheme have good stability properties for a particular equation (e.g., our particular A of (21.2))? And for what Δt will the scheme be stable? (The latter is particularly important for explicit schemes.)

To address these questions we again apply modal analysis but now to our discrete equations (21.14). In particular, we write

$$\tilde{w}^j = \tilde{z}_1^j\chi^1 + \tilde{z}_2^j\chi^2\ , \tag{21.15}$$

where χ^1 and χ^2 are the eigenvectors of A as derived in the previous section. We now insert (21.15) into (21.14) and multiply by $(\psi^1)^{\mathrm{H}}$ and $(\psi^2)^{\mathrm{H}}$ — just as in the previous section — to obtain

$$\tilde{z}_1^j = \tilde{z}_1^{j-1} + \frac{\lambda_1 \Delta t}{2}\left(\tilde{z}_1^j + \tilde{z}_1^{j-1}\right) + (\psi^1)^{\mathrm{H}}\,\frac{\Delta t}{2}\left(F^j + F^{j-1}\right) , \tag{21.16}$$

$$\tilde{z}_2^j = \tilde{z}_2^{j-1} + \frac{\lambda_2 \Delta t}{2}\left(\tilde{z}_2^j + \tilde{z}_2^{j-1}\right) + (\psi^2)^{\mathrm{H}}\,\frac{\Delta t}{2}\left(F^j + F^{j-1}\right) , \tag{21.17}$$

with corresponding initial conditions (which are not relevant to our current discussion).

We now recall that for the model problem

$$\frac{du}{dt} = \lambda u + f , \tag{21.18}$$

analogous to (21.10), we arrive at the Crank-Nicolson scheme

$$\tilde{u}^j = \tilde{u}^{j-1} + \frac{\lambda \Delta t}{2}(\tilde{u}^j + \tilde{u}^{j-1}) + \frac{\Delta t}{2}(f^j + f^{j-1}) , \tag{21.19}$$

analogous to (21.16). Working backwards, for (21.19) and hence (21.16) to be a stable approximation to (21.18) and hence (21.10), we must require $\lambda \Delta t$, and hence $\lambda_1 \Delta t$, to reside in the Crank-Nicolson absolute stability region depicted in Figure 21.12(a). Put more bluntly, we know that the difference equation (21.16) will blow up — and hence also (21.14) by virtue of (21.15) — if $\lambda_1 \Delta t$ is not in the unshaded region of Figure 21.12(a). By similar arguments, $\lambda_2 \Delta t$ must also lie in the unshaded region of Figure 21.12(a). In this case, we know that both λ_1 and λ_2 — for our *particular* equation, that is, for our *particular* matrix A (which determines the eigenvalues λ_1, λ_2) — are in the left-hand plane, and hence in the Crank-Nicolson absolute stability region; thus Crank-Nicolson is unconditionally stable — stable for all Δt — for our particular equation and will converge as $O(\Delta t^2)$ as $\Delta t \to 0$.

We emphasize that the numerical procedure is given by (21.14) , and *not* by (21.16), (21.17). The modal decomposition is just for the purposes of understanding and analysis — to determine if a scheme is stable and if so for what values of Δt. (For a 2×2 matrix A the full modal decomposition is simple. But for larger systems, as we will consider in the next section, the full modal decomposition is very expensive. Hence we prefer to directly discretize the original equation, as in (21.14). This direct approach is also more general, for example for treatment of nonlinear problems.) It follows that Δt in (21.16) and (21.17) are the *same* — both originate in the equation (21.14). We discuss this further below in the context of stiff equations.

General Recipe

We now consider a general system of $n = 2$ ODEs given by

$$\begin{aligned} \frac{dw}{dt} &= Aw + F , \\ w(0) &= w_0 , \end{aligned} \tag{21.20}$$

where $w \in \mathbb{R}^2$, $A \in \mathbb{R}^{2\times 2}$ (a 2×2 matrix), $F \in \mathbb{R}^2$, and $w_0 \in \mathbb{R}^2$. We next discretize (21.20) by any of the schemes developed earlier for the scalar equation

$$\frac{du}{dt} = g(u,t)$$

simply by substituting w for u and $Aw + F$ for $g(u, t)$. We shall denote the scheme by $\mathbb{S}$ and the associated absolute stability region by $\mathcal{R}_{\mathbb{S}}$. Recall that $\mathcal{R}_{\mathbb{S}}$ is the subset of the complex plane which contains all $\lambda\Delta t$ for which the scheme $\mathbb{S}$ applied to $g(u, t) = \lambda u$ is absolutely stable.

For example, if we apply the Euler Forward scheme $\mathbb{S}$ we obtain

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t(A\tilde{w}^{j-1} + F^{j-1}) \ , \tag{21.21}$$

whereas Euler Backward as $\mathbb{S}$ yields

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t(A\tilde{w}^{j} + F^{j}) \ , \tag{21.22}$$

and Crank-Nicolson as $\mathbb{S}$ gives

$$\tilde{w}^j = \tilde{w}^{j-1} + \frac{\Delta t}{2}(A\tilde{w}^{j} + A\tilde{w}^{j-1}) + \frac{\Delta t}{2}(F^j + F^{j-1}) \ . \tag{21.23}$$

A multistep scheme such as AB2 as $\mathbb{S}$ gives

$$\tilde{w}^j = \tilde{w}^{j-1} + \Delta t\left(\frac{3}{2}\,A\tilde{w}^{j-1} - \frac{1}{2}\,A\tilde{w}^{j-2}\right) + \Delta t\left(\frac{3}{2}\,F^{j-1} - \frac{1}{2}\,F^{j-2}\right) \ . \tag{21.24}$$

The stability diagrams for these four schemes, $\mathcal{R}_{\mathbb{S}}$, are given by Figure 21.9, Figure 21.7, Figure 21.12(a), and Figure 21.11(b), respectively.

We next assume that we can calculate the two eigenvalues of A, λ_1, and λ_2. A particular Δt will lead to a *stable* scheme if and only if the two points $\lambda_1\Delta t$ and $\lambda_2\Delta t$ *both* lie inside $\mathcal{R}_{\mathbb{S}}$. If either or both of the two points $\lambda_1\Delta t$ or $\lambda_2\Delta t$ lie outside $\mathcal{R}_{\mathbb{S}}$, then we must decrease Δt until both $\lambda_1\Delta t$ and $\lambda_2\Delta t$ lie inside $\mathcal{R}_{\mathbb{S}}$. The critical time step, Δt_{cr}, is defined to be the *largest* Δt for which the two *rays* $[0, \lambda_1\Delta t]$, $[0, \lambda_2\Delta t]$, *both* lie within $\mathcal{R}_{\mathbb{S}}$; Δt_{cr} will depend on the shape and size of $\mathcal{R}_{\mathbb{S}}$ and the "orientation" of the two rays $[0, \lambda_1\Delta t]$, $[0, \lambda_2\Delta t]$.

We can derive Δt_{cr} in a slightly different fashion. We first define $\widehat{\Delta t}_1$ to be the largest Δt such that the ray $[0, \lambda_1\Delta t]$ is in $\mathcal{R}_{\mathbb{S}}$; we next define $\widehat{\Delta t}_2$ to be the largest Δt such that the ray $[0, \lambda_2\Delta t]$ is in $\mathcal{R}_{\mathbb{S}}$. We can then deduce that $\Delta t_{\text{cr}} = \min(\widehat{\Delta t}_1, \widehat{\Delta t}_2)$. *In particular, we note that if* $\Delta t > \Delta t_{\text{cr}}$ *then one of the two modes — and hence the entire solution — will explode.* We can also see here again the difficulty with stiff equations in which λ_1 and λ_2 are very different: $\widehat{\Delta t}_1$ may be (say) much larger than $\widehat{\Delta t}_2$, but $\widehat{\Delta t}_2$ will dictate Δt and thus force us to take many time steps — many more than required to resolve the slower mode (smaller $|\lambda_1|$ associated with slower decay or slower oscillation) which is often the behavior of interest.

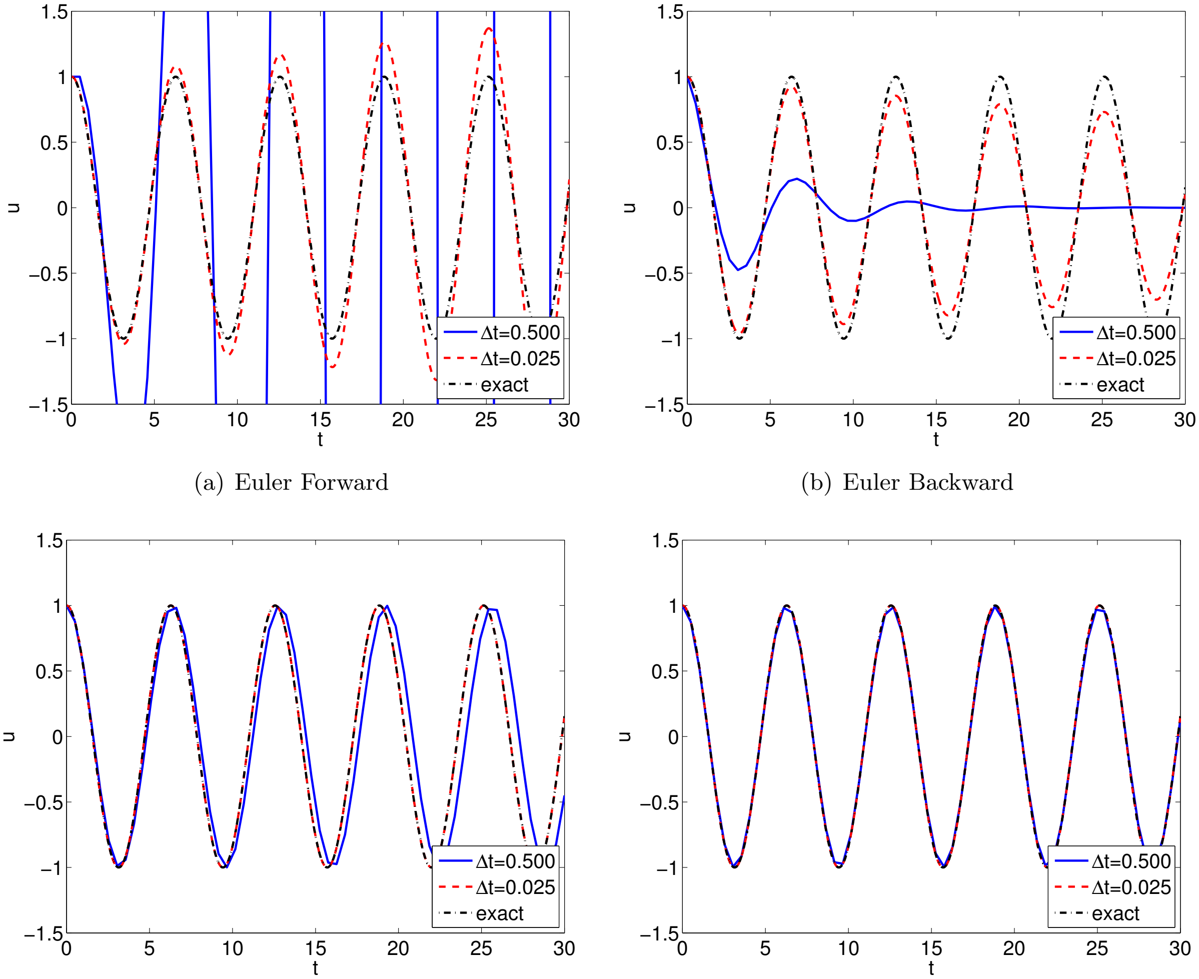
In the above we assumed, as is almost always the case, that the λ are in the left-hand plane. For any λ which are in the right-hand plane, our condition is flipped: we now must make sure that the $\lambda\Delta t$ are *not* in the absolute stability region in order to obtain the desired growing (unstable) solutions.

Let us close this section with two examples.

Example 21.3.1 Undamped spring-mass system

In this example, we revisit the undamped spring-mass system considered in the previous section. The two eigenvalues of A are $\lambda_1 = i\omega_n$ and $\lambda_2 = i\omega_n$; without loss of generality, we set $\omega_n = 1.0$. We will consider application of several different numerical integration schemes to the problem; for each integrator, we assess its applicability based on theory (by appealing to the absolute stability diagram) and verify our assessment through numerical experiments.

(*i*) Euler Forward is a poor choice since both $\lambda_1\Delta t$ and $\lambda_2\Delta t$ are outside $\mathcal{R}_{\mathbb{S}=\text{EF}}$ for all Δt. The result of numerical experiment, shown in Figure 21.20(a), confirms that the amplitude of the oscillation grows for both $\Delta t = 0.5$ and $\Delta t = 0.025$; the smaller time step results in a smaller (artificial) amplification.



(a) Euler Forward

(b) Euler Backward

(c) Crank-Nicolson

(d) Four-stage Runge-Kutta

Figure 21.20: Comparison of numerical integration schemes for an undamped spring-mass system with $\omega_n = 1.0$.

(*ii*) Euler Backward is also a poor choice since $\lambda_1\Delta t$ and $\lambda_2\Delta t$ are in the *interior* of $\mathcal{R}_{\mathbb{S}=\mathrm{EB}}$ for all Δt and hence the discrete solution will decay even though the exact solution is a *non-decaying* oscillation. Figure 21.20(b) confirms the assessment.

(*iii*) Crank-Nicolson is a very good choice since $\lambda_1\Delta t \in \mathcal{R}_{\mathbb{S}=\mathrm{CN}}$, $\lambda_2\Delta t \in \mathcal{R}_{\mathbb{S}=\mathrm{CN}}$ for all Δt, and furthermore $\lambda_1\Delta t$, $\lambda_2\Delta t$ lie on the *boundary* of $\mathcal{R}_{\mathbb{S}=\mathrm{CN}}$ and hence the discrete solution, just as the exact solution, will not decay. Figure 21.20(c) confirms that Crank-Nicolson preserves the amplitude of the response regardless of the choice of Δt; however, the $\Delta t = 0.5$ case results in a noticeable phase error.

(*iv*) Four-stage Runge-Kutta (RK4) is a reasonably good choice since $\lambda_1\Delta t$ and $\lambda_2\Delta t$ lie close to the boundary of $\mathcal{R}_{\mathbb{S}=\mathrm{RK4}}$ for $|\lambda_i\Delta t| \lesssim 1$. Figure 21.20(d) shows that, for the problem considered, RK4 excels at not only preserving the amplitude of the oscillation but also at attaining the correct phase.

Note in the above analysis the absolute stability diagram serves not only to determine stability but also the nature of the discrete solution as regards growth, *or* decay, *or* even neutral stability — no growth or decay. (The latter does not imply that the discrete solution is exact, since in addition to amplitude errors there are also phase errors. Our Crank-Nicolson result, shown in Figure 21.20(c), in particular demonstrate the presence of phase errors in the absence of amplitude errors.)

Example 21.3.2 Overdamped spring-mass-damper system: a stiff system of ODEs
In our second example, we consider a (very) overdamped spring-mass-damper system with $\omega_n = 1.0$ and $\zeta = 100$. The eigenvalues associated with the system are

$$\lambda_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1} = -0.01$$
$$\lambda_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1} = -99.99\,.$$

As before, we perturb the system by a unit initial displacement. The slow mode with $\lambda_1 = -0.01$ dictates the response of the system. However, for conditionally stable schemes, the stability is governed by the fast mode with $\lambda_2 = -99.99$. We again consider four different time integrators: two explicit and two implicit.

(*i*) Euler Forward is stable for $\Delta t \lesssim 0.02$ (i.e. $\Delta t_{\mathrm{cr}} = 2/|\lambda_2|$). Figure 21.21(a) shows that the scheme accurately tracks the (rather benign) exact solution for $\Delta t = 0.02$, but becomes unstable and diverges exponentially for $\Delta t = 0.0201$. Thus, the maximum time step is limited not by the ability to approximate the system response (dictated by λ_1) but rather by stability (dictated by λ_2). In other words, even though the system response is benign, we cannot use large time steps to save on computational cost.

(*ii*) Similar to the Euler Forward case, the four-stage Runge-Kutta (RK4) scheme exhibits an exponentially diverging behavior for $\Delta t > \Delta t_{\mathrm{cr}} \approx 0.028$, as shown in Figure 21.21(b). The maximum time step is again limited by stability.

(*iii*) Euler Backward is unconditionally stable, and thus the choice of the time step is dictated by the ability to approximate the system response, which is dictated by λ_1. Figure 21.21(c) shows that Euler Backward in fact produces a good approximation even for a time step as large as $\Delta t = 5.0$ since the system response is rather slow.

(*iv*) Crank-Nicolson is also unconditionally stable. For the same set of time steps, Crank-Nicolson produces a more accurate approximation than Euler Backward, as shown in Figure 21.21(d), due to its higher-order accuracy.

In the above comparison, the unconditionally stable schemes required many fewer time steps (and hence much less computational effort) than conditionally stable schemes. For instance, Crank-Nicolson with $\Delta t = 5.0$ requires approximately 200 times fewer time steps than the RK4 scheme (with a stable choice of the time step). More importantly, as the shortest time scale (i.e. the largest eigenvalue) dictates stability, conditionally stable schemes do not allow the user to use large time steps *even if the fast modes are of no interest to the user*. As mentioned previously, stiff systems are ubiquitous in engineering, and engineers are often not interested in the smallest time scale present in the system. (Recall the example of the time scale associated with the dynamics of a passenger jet and that associated with turbulent eddies; engineers are often only interested in characterizing the dynamics of the aircraft, not the eddies.) In these situations, unconditionally stable schemes allow users to choose an appropriate time step independent of stability limitations.

·

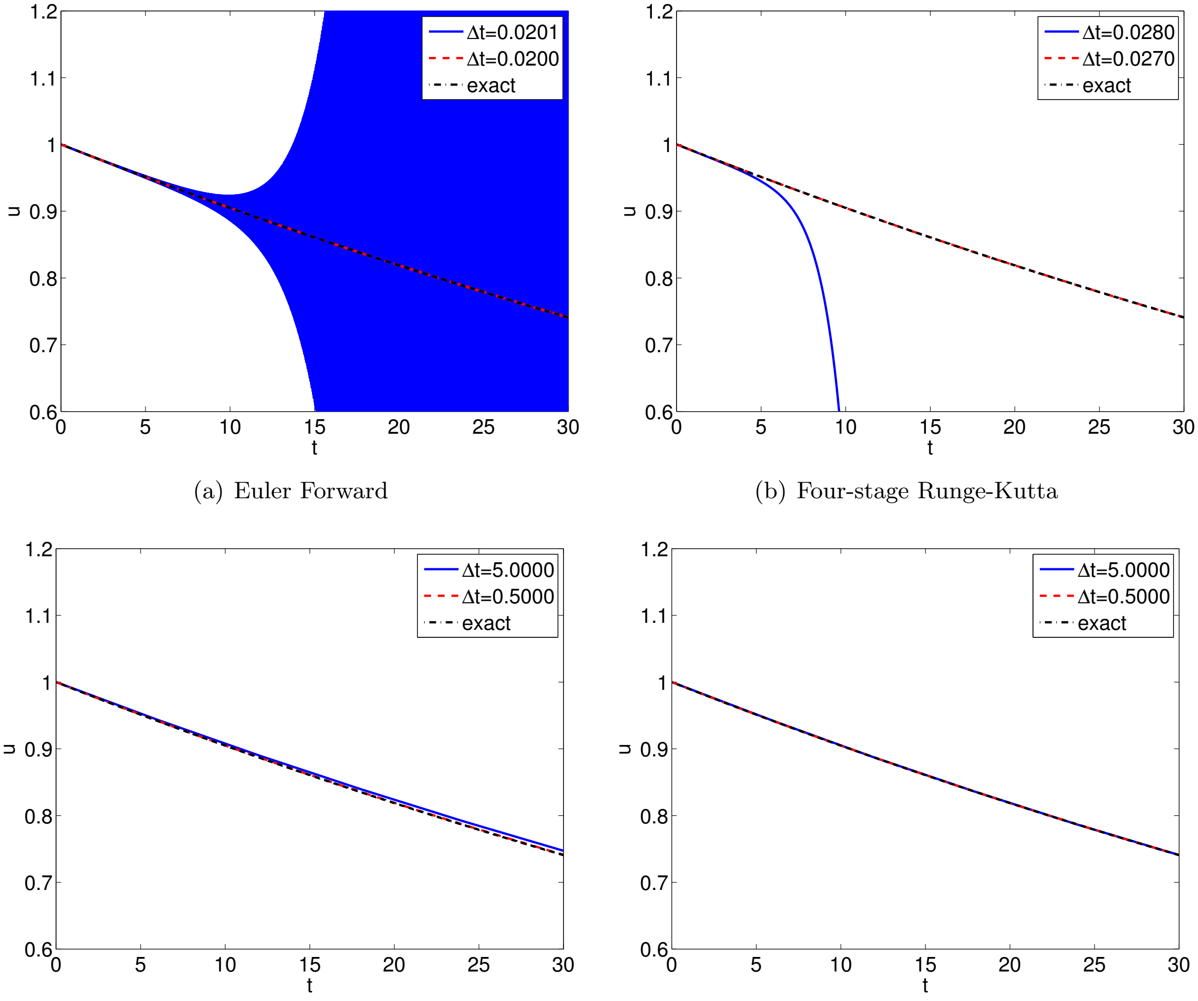
In closing, it is clear even from these simple examples that a general purpose *explicit* scheme would ideally include some part of both the negative real axis *and* the imaginary axis. Schemes that exhibit this behavior include AB3 and RK4. Of these two schemes, RK4 is often preferred due to a large stability region; also RK4, a multi-*stage* method, does not suffer from the start-up issues that sometimes complicate multi-step techniques.

21.4 IVPs: System of n Linear ODEs

We consider here for simplicity a particular family of problems: $n/2$ coupled oscillators. This family of systems can be described by the set of equations.

$$\begin{aligned}
\frac{d^2u^{(1)}}{dt^2} &= g^{(1)}\left(\frac{du^{(j)}}{dt}, u^{(j)},\ 1 \le j \le n/2\right) + f^{(1)}(t)\ ,\\
\frac{d^2u^{(2)}}{dt^2} &= g^{(2)}\left(\frac{du^{(j)}}{dt}, u^{(j)},\ 1 \le j \le n/2\right) + f^{(2)}(t)\ ,\\
&\ \ \vdots\\
\frac{d^2u^{(n/2)}}{dt^2} &= g^{(n/2)}\left(\frac{du^{(j)}}{dt}, u^{(j)},\ 1 \le j \le n/2\right) + f^{(n/2)}(t)\ ,
\end{aligned}$$

where $g^{(k)}$ is assumed to be a *linear* function of all its arguments.



(a) Euler Forward

(b) Four-stage Runge-Kutta

(c) Euler Backward

(d) Crank-Nicolson

Figure 21.21: Comparison of numerical integration schemes for an overdamped spring-mass-damper system with $\omega_n = 1.0$ and $\zeta = 50$. Note that the time step used for the explicit schemes are different from those for the implicit schemes.

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

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

We can then write our system — using the fact that g is linear in its arguments — as

$$\begin{aligned} \frac{dw}{dt} &= Aw + F \\ w(0) &= w_0 \end{aligned} \tag{21.25}$$

where g determines A, F is given by $\begin{pmatrix} 0 & f^{(1)}(t) & 0 & f^{(2)}(t) & \ldots & 0 & f^{(n/2)}(t) \end{pmatrix}^{\mathrm{T}}$, and

$$w_0 = \begin{pmatrix} u^{(1)}(0) & \dfrac{du^{(1)}}{dt}(0) & u^{(2)}(0) & \dfrac{du^{(2)}}{dt}(0) & \ldots & u^{(n/2)}(0) & \dfrac{du^{(n/2)}}{dt}(0) \end{pmatrix}^{\mathrm{T}}\,.$$

We have now reduced our problem to an abstract form identical to (21.20) and hence we may apply any scheme $\mathbb{S}$ to (21.25) in the same fashion as to (21.20).

For example, Euler Forward, Euler Backward, Crank-Nicolson, and AB2 applied to (21.25) will take the same form (21.21), (21.22), (21.23), (21.24), respectively, except that now $w \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $F \in \mathbb{R}^n$, $w_0 \in \mathbb{R}^n$ are given in (21.25), where $n/2$, the number of oscillators (or masses) in our system, is no longer restricted to $n/2 = 1$ (i.e., $n = 2$). We can similarly apply AB3 or BD2 or RK4.

Our stability criterion is also readily extended. We first note that A will now have in general n eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_n$. (In certain cases multiple eigenvalues can create difficulties; we do not consider these typically rather rare cases here.) Our stability condition is then simply stated: a time step Δt will lead to stable behavior if and only if $\lambda_i \Delta t$ is in $\mathcal{R}_{\mathbb{S}}$ for all i, $1 \le i \le n$. If this condition is not satisfied then there will be one (or more) modes which will explode, taking with it (or them) the entire solution. (For certain very special initial conditions — in which the w_0 is chosen such that all of the dangerous modes are initially *exactly* zero — this blow-up could be avoided in *infinite* precision; but in finite precision we would still be doomed.) For explicit schemes, Δt_{cr} is the *largest* time step such that *all* the rays $[0, \lambda_i \Delta t]$, $1 \le i \le n$, lie within $\mathcal{R}_{\mathbb{S}}$.

There are certainly computational difficulties that arise for large n that are not an issue for $n = 2$ (or small n). First, for implicit schemes, the necessary division — *solution* rather than evaluation of matrix-vector equations — will become considerably more expensive. Second, for explicit schemes, determination of Δt_{cr}, or a bound $\Delta t_{\mathrm{cr}}^{\mathrm{conservative}}$ such that $\Delta t_{\mathrm{cr}}^{\mathrm{conservative}} \approx \Delta t_{\mathrm{cr}}$ and $\Delta t_{\mathrm{cr}}^{\mathrm{conservative}} \le \Delta t_{\mathrm{cr}}$, can be difficult. As already mentioned, the full modal decomposition can be expensive. Fortunately, in order to determine Δt_{cr}, we often only need as estimate for say the most negative real eigenvalue, or the largest (in magnitude) imaginary eigenvalue; these extreme eigenvalues can often be estimated relatively efficiently.

Finally, we note that in practice often adaptive schemes are used in which stability and accuracy are monitored and Δt modified appropriately. These methods can also address nonlinear problems — in which g no longer depends linearly on its arguments.

Chapter 22

Boundary Value Problems

Chapter 23

Partial Differential Equations