

A New Second-order Integration Algorithm for Simulating Mechanical Dynamic Systems

R.M. Howe

Department of Aerospace Engineering, The University of Michigan, Ann Arbor, Michigan
and
Applied Dynamics International, Ann Arbor, Michigan

ABSTRACT

A new integration algorithm which has the simplicity of Euler integration but exhibits second-order accuracy is described. In fixed-step numerical integration of differential equations for mechanical dynamic systems the method represents displacement and acceleration variables at integer step times and velocity variables at half-integer step times. Asymptotic accuracy of the algorithm is twice that of trapezoidal integration and ten times that of second-order Adams-Bashforth integration. The algorithm is also compatible with real-time inputs when used for a real-time simulation. It can be used to produce simulation outputs at double the integration frame rate, i.e., at both half-integer and integer frame times, even though it requires only one evaluation of state-variable derivatives per integration step. The new algorithm is shown to be especially effective in the simulation of lightly-damped structural modes. Both time-domain and frequency-domain accuracy comparisons with traditional integration methods are presented. Stability of the new algorithm is also examined.

1. Introduction

In the simulation of mechanical dynamic systems described by ordinary differential equations the required dynamic accuracy is often modest, especially when the real-time computation is utilized as part of a hardware-in-the-loop simulation. Accuracies ranging between 0.1 and 1 percent are considered adequate in many cases. For this reason, lower-order numerical integration algorithms are often employed. Also, fixed integration time steps are invariably used in real-time simulations in order to assure that the simulation outputs for each integration step occur at a fixed rate that can be synchronized with real time. In fact, the Adams-Bashforth second-order predictor algorithm, hereafter referred to as AB-2, is perhaps the most widely used method for real-time simulation.

In this paper we consider a modified form of Euler integration which is well suited to the dynamic simulation of mechanical systems. It is especially effective in the simulation of systems with lightly-damped oscillatory modes, such as flexible structures. The method has the simplicity of conventional Euler integration but exhibits dynamic errors that are second order rather than first order in the integration step size h . Also, the dynamic error coefficients associated with the method are smaller than those for any other second-order method. In the next section we introduce the basic concept behind the modified Euler method as it is used in the dynamic simulation of mechanical systems. This is followed by a discussion of dynamic error

measures with emphasis on the frequency domain. Several example simulations are then introduced to demonstrate the accuracy improvement achieved when using the modified Euler method instead of conventional algorithms. The stability boundaries for different versions of the modified Euler method are also compared with those for conventional methods.

2. The Modified-Euler Method

The simulation of mechanical dynamic systems normally requires the integration of a time-dependent acceleration $A(t)$ to obtain a velocity V , followed by a second integration to obtain a displacement D . This is illustrated diagrammatically in Figure 1. One method for implementing the required integrations is to use the forward Euler formula for the first integration and the backward Euler formula for the second integration. The required difference equations are the following:

$$V_{n+1} = V_n + hA_n \quad , \quad D_{n+1} = D_n + hV_{n+1} \quad (1)$$

Here h is the integration step size and A_n , V_n and D_n represent the respective variables at the time $t = nh$, where n is an integer. Eq. (1) has been used in real-time simulation to achieve dynamic accuracy improvement over that obtained when using the forward Euler formula for both integrations. In Eq. (1) the first-order error associated with the forward Euler formula cancels the equal and opposite first-order error associated with the backward Euler formula. As a result the displacement D exhibits second-order accuracy with respect to the input acceleration A .

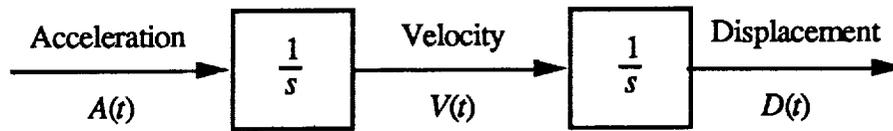


Figure 1. Paired integration to obtain velocity and displacement from acceleration.

Both integrations become second order if we consider the velocity to be represented at a half-integer frame. In this case the difference equations become

$$V_{n+1/2} = V_{n-1/2} + hA_n \quad , \quad D_{n+1} = D_n + hV_{n+1/2} \quad (2)$$

The acceleration A will, of course, usually be a function of both velocity V and displacement D , as well as an explicit time-dependent input $U(t)$. In this case we can write the system state equations as

$$\dot{V} = A[D, V, U(t)] \quad , \quad \dot{D} = V \quad (3)$$

where in general the variables will be vectors rather than scalars. In Eq. (3) we see that the acceleration A_n at the n th frame depends on the velocity V_n at the n th frame, which is not available in the half-integer representation for V as utilized in Eq. (2). The best we can do is to employ an estimate \hat{V}_n for V_n based on half-integer values $V_{n-1/2}$, $V_{n-3/2}$, etc. Then the modified Euler difference equations are given by

$$V_{n+1/2} = V_{n-1/2} + hA(D_n, \hat{V}_n, U_n) , \quad D_{n+1} = D_n + hV_{n+1/2} \quad (4)$$

Table 1 lists some possible candidate formulas for estimating V_n . In the first formula in Table 1 we let $\hat{V}_n = V_{n-1/2}$. This is equivalent to using conventional Euler integration rather than modified Euler integration for the V dependent portion of $A(D, V, U)$, with the corresponding dynamic error proportional to h . The second formula for \hat{V}_n uses a linear extrapolation based on $V_{n-1/2}$ and $V_{n-3/2}$. It is equivalent to using AB-2 integration for the V dependent portion of A , with the corresponding dynamic error proportional to h^2 . In the third formula \hat{V}_n is derived from averaging $V_{n+1/2}$ and $V_{n-1/2}$. It is equivalent to trapezoidal integration for the V dependent portion of A and represents an implicit formulation, since $V_{n+1/2}$ now appears on both sides of the left equation in (4). Later in this section we will see how this can be turned into an explicit formulation in many cases. Finally, the last formula in Table 1 uses a second-order predictor integration method to obtain \hat{V}_n from $V_{n-1/2}$ and the derivatives \dot{V}_{n-1} and \dot{V}_{n-2} . It produces a local truncation error in \hat{V}_n proportional to h^3 and therefore permits the full accuracy of the modified Euler method to be realized.

Table 1. Methods for Estimating \hat{V}_n in $A(D_n, \hat{V}_n, U_n)$

Euler	$\hat{V}_n = V_{n-1/2}$
AB-2	$\hat{V}_n = \frac{3}{2}V_{n-1/2} - \frac{1}{2}V_{n-3/2}$
Trapezoidal	$\hat{V}_n = \frac{V_{n+1/2} + V_{n-1/2}}{2}$
Predictor Integrator	$V_n = V_{n-1/2} + h(\frac{7}{8}\dot{V}_{n-1} - \frac{3}{8}\dot{V}_{n-2})$

Before considering some examples of the application of the modified-Euler methods described here, we consider some dynamic error measures for examining comparative accuracy of different integration algorithms.

3. Integrator Error Measures

Consider the solution of the state equation $dy/dt = f(t)$ using a numerical integration formula for y_{n+1} in terms of y_n and the derivative f . Furthermore, let $y[(n+1)h]$ and $y[nh]$ represent the exact solution of the continuous system at the times $t = (n+1)h$ and nh , respectively. Then we can then write

$$y_{n+1} - y_n \equiv y[(n+1)h] - y[nh] - e_I f_n^{(k)} h^{k+1} \quad (5)$$

Here the term $-e_I f_n^{(k)} h^{k+1}$ represents the local truncation error associated with the integration method of order k and $f_n^{(k)}$ is the k th time derivative of f at $t = nh$ [1]. For example, $k = 1$ and $e_I = 1/2$ for Euler integration; for AB-2 integration $k = 2$ and $e_I = 5/12$. We now take the Z transform of Eq. (5) and divide by $z - 1$ to obtain

$$Y^*(z) \equiv Y_{ref}^*(z) - \frac{e_I h^{k+1} F^{(k)*}(z)}{z - 1} \quad (6)$$

Here $Y_{ref}^*(z)$ is the Z transform of the exact solution, $y[nh]$. Next we consider the case of sinusoidal data sequences by replacing z with $e^{j\omega h}$. We also note that $F^{(k)*}(e^{j\omega h}) = (j\omega)^k F^*(e^{j\omega h})$, i.e., the Fourier transform of the k th derivative of a function is equal to the Fourier transform of the function multiplied by $(j\omega)^k$. After dividing the resulting expression by F^* , we have

$$\frac{Y^*(e^{j\omega h})}{F^*(e^{j\omega h})} \equiv \frac{Y_{ref}^*(e^{j\omega h})}{F^*(e^{j\omega h})} - \frac{e_I h (j\omega h)^k}{e^{j\omega h} - 1} \quad (7)$$

The term Y^*/F^* is simply the sinusoidal transfer function, $H_I^*(e^{j\omega h})$, of the numerical integrator. The term $Y_{ref}^*/F^* = 1/j\omega$, the transfer function of an ideal integrator. If we now approximate $e^{j\omega h} - 1$ by $j\omega h$, Eq. (7) becomes the following:

$$H_I^*(e^{j\omega h}) \equiv \frac{1 - e_I (j\omega h)^k}{j\omega} \equiv \frac{1}{j\omega [1 + e_I (j\omega h)^k]}, \quad \omega h \ll 1 \quad (8)$$

Here e_I is the integrator error coefficient and k is the algorithm order. To illustrate the application of our integrator transfer function model, we consider the simulation of a linearized dynamic system with transfer function $H(s)$. For the case of sinusoidal inputs of frequency ω , the transfer function becomes $H(j\omega)$. When the continuous system is simulated with a single-pass integration method, the sinusoidal transfer function of the digital simulation is simply given by $H(1/H_I^*)$, where H_I^* is the transfer function of the digital integrator. For $\omega h \ll 1$, $1/H_I^*$ can be approximated by $j\omega [1 + e_I (j\omega h)^k]$ in accordance with the integrator model of Eq. (8). Thus the formula for the transfer function of the digital system in simulating the linear system with transfer function $H(s)$ is given by

$$H^*(e^{j\omega h}) = H(1/H_I^*) \equiv H\{j\omega [1 + e_I (j\omega h)^k]\}, \quad \omega h \ll 1 \quad (9)$$

For example, consider a first-order linear system with eigenvalue λ and transfer function given by $H(s) = 1/(s - \lambda)$. The transfer function for sinusoidal inputs becomes

$$H(j\omega) = \frac{1}{j\omega - \lambda} \quad (10)$$

Then the digital system transfer function for sinusoidal input data sequences is given approximately by

$$H^*(e^{j\omega h}) \cong \frac{1}{j\omega[1 + e_1(j\omega h)^k] - \lambda}, \quad j\omega \ll 1 \quad (11)$$

We note that the characteristic root (eigenvalue) λ for the continuous system is given by the value of $j\omega$ in Eq. (10) which makes the denominator vanish. It follows that the equivalent characteristic root λ^* for the digital system is given by the value of $j\omega$ which makes the denominator of Eq. (11) vanish. Replacing $j\omega$ by λ^* in Eq. (11) and setting the denominator equal to zero, we can write

$$\lambda^* = \lambda - \lambda^* e_1 (\lambda^* h)^k$$

For $|\lambda h| \ll 1$, $\lambda^* \cong \lambda$ to order h^k . Then we can replace λ^* by λ on the right side of the equation and obtain

$$e_\lambda = \frac{\lambda^* - \lambda}{\lambda} \cong -e_1 (\lambda h)^k, \quad |\lambda h| \ll 1 \quad (12)$$

Here e_λ represents the fractional error in the digital system characteristic root. We recall that the transfer function for any finite order linear system with distinct roots can be represented as the sum of first-order transfer functions of the form $1/(s - \lambda)$, where the characteristic roots may be real or complex. It follows that Eq. (12) can be used to estimate the error in each characteristic root in the digital system simulation of any order linear system.

From the digital transfer function formula in Eq. (11) we can write

$$H^*(e^{j\omega h}) \cong \frac{1}{(j\omega - \lambda) \left[1 + \frac{e_1 j\omega (j\omega h)^k}{j\omega - \lambda} \right]} = \frac{H(j\omega)}{1 + \frac{e_1 j\omega (j\omega h)^k}{j\omega - \lambda}}$$

from which

$$\frac{H^*(e^{j\omega h}) - H(j\omega)}{H(j\omega)} \cong -\frac{e_1 j\omega (j\omega h)^k}{j\omega - \lambda}, \quad \omega h \ll 1 \quad (13)$$

Here $(H^* - H)/H$ represents the fractional error in digital system transfer function. For $\omega h \ll 1$ it is evident that this fractional error will be small in magnitude compared with unity. In this case it can be shown that the real part of $(H^* - H)/H$ is approximately equal to the fractional gain error

of the digital transfer function and the imaginary part is approximately equal to the phase error [2]. We note that the transfer function for any finite-order linear system can be written as the product of individual pole and zero factors of the form $(s - \lambda)$, where again λ can be either real or complex. It is then straightforward to show that the fractional error in the overall digital transfer function is approximately the sum of the individual errors given by Eq. (13) for each factor [2]. It follows that both gain and phase errors of the overall digital system transfer function for sinusoidal inputs are proportional to $e_I(j\omega h)^k$.

Thus for single-pass integration methods Eq. (12) and (13) represent simple approximate formulas for both characteristic root and transfer function errors. For a given integration algorithm the errors are directly proportional to the integrator error coefficient e_I for that algorithm. Table 2 lists e_I and k for the algorithms considered in this paper, including the modified-Euler method, which has the smallest error coefficient ($e_I = 1/24$).

Table 2. Error Coefficients for Integration Methods

$$\text{Integrator transfer function} = H_I^*(e^{j\omega h}) \cong \frac{1}{j\omega[1 + e_I(j\omega h)^k]}, \quad \omega h \ll 1$$

	e_I	k
Euler	$\frac{1}{2}$	1
AB-2	$\frac{5}{12}$	2
Trapezoidal	$-\frac{1}{12}$	2
Modified Euler	$\frac{1}{24}$	2

All of the above algorithms in Table 2 are explicit except trapezoidal, which is implicit. An explicit method with the same asymptotic accuracy can, however, be realized with the two-pass Adams-Moulton (AM-2) algorithm. In this method the first pass employs AB-2 integration to obtain an estimate \hat{y}_{n+1} of the next state. This is then used in the trapezoidal formula to compute the corrected y_{n+1} . The local truncation error associated with \hat{y}_{n+1} is of order h^3 , which ensures that the asymptotic accuracy of order h^2 for the corrected y_{n+1} will be the same as that for implicit trapezoidal integration.

4. Specific Examples

We now turn to some specific examples to compare the accuracy of modified Euler integration with traditional algorithms of second order. We consider first a simple linear dependence of the acceleration A in Eq. (3) on the displacement D and velocity V . This leads to

the following state equations, which for convenience have been written in terms of the undamped natural frequency ω_n and damping ratio ζ of the second-order system:

$$\dot{V} = \omega_n^2(U_n - D_n) - 2\zeta\omega_n V, \quad \dot{D} = V \quad (14)$$

From Eqs. (2) and (4), with the trapezoidal formula from Table 1 used for \hat{V}_{n+1} , we obtain the following difference equations for the modified Euler formulation:

$$V_{n+1/2} = V_{n-1/2} + \omega_n^2 h(U_n - D_n) - \zeta\omega_n(V_{n+1/2} + V_{n-1/2}), \quad D_{n+1} = D_n + hV_{n+1/2}$$

After solving the first equation for $V_{n+1/2}$, we have the following explicit equations:

$$V_{n+1/2} = C_1 V_{n-1/2} + C_2(U_n - D_n), \quad D_{n+1} = D_n + hV_{n+1/2} \quad (15)$$

where

$$C_1 = \frac{1 - \zeta\omega_n h}{1 + \zeta\omega_n h}, \quad C_2 = \frac{\omega_n^2 h}{1 + \zeta\omega_n h} \quad (16)$$

Here the constants C_1 and C_2 can be precomputed. From Eq. (15) it follows that the ongoing simulation run then requires only 3 adds and 3 multiples per integration step.

Figure 2 shows plots of the solution error when using modified Euler integration to compute the response of a second-order system with $\zeta = 0.25$ to a unit step input. The initial conditions are given by $x(0) = y(0) = 0$. Shown in the figure are error plots for three of the damping methods listed in Table 1, including the trapezoidal damping used to derive Eqs. (15) and (16). For comparison Figure 2 also shows the step-response errors when AB-2 integration is used. In all cases the step size is given by $\omega_n h = 0.25$. The startup problem associated with AB-2 integration (the initial states at $t = -h$ are not specified) is solved by using Euler integration for the first step. In the case of modified Euler integration the first step which computes $y_{1/2}$ from y_0 uses a step equal to $h/2$. The figure clearly shows the superior accuracy of the modified Euler method, with the scheme using second-order predictor integration to estimate \hat{y}_n producing the smallest errors.

The second example considered in this section is the simulation of the full nonlinear flight equations of an aircraft. Since the largest characteristic roots for the rigid airframe are normally those associated with the short-period pitching motion, we will only consider symmetric flight, i.e., the longitudinal equations of motion, in our example simulation. The conclusions regarding dynamic errors can be safely extrapolated to the full six-degree-of-freedom case. For this simulation the translational equations of motion are written with respect to flight-path axes, while the rotational equations of motion are written with respect to body axes [3]. Then the velocity state variables become total aircraft velocity V_p , angle of attack α , and pitch rate Q . The displacement state variables are altitude H , pitch angle Θ , and horizontal distance X . The velocity state equations are given by

$$\dot{V}_p = \frac{F_{wx}}{m}, \quad \dot{\alpha} = Q + \frac{F_{wz}}{mV_p}, \quad \dot{Q} = \frac{M}{I_{yy}} \quad (17)$$

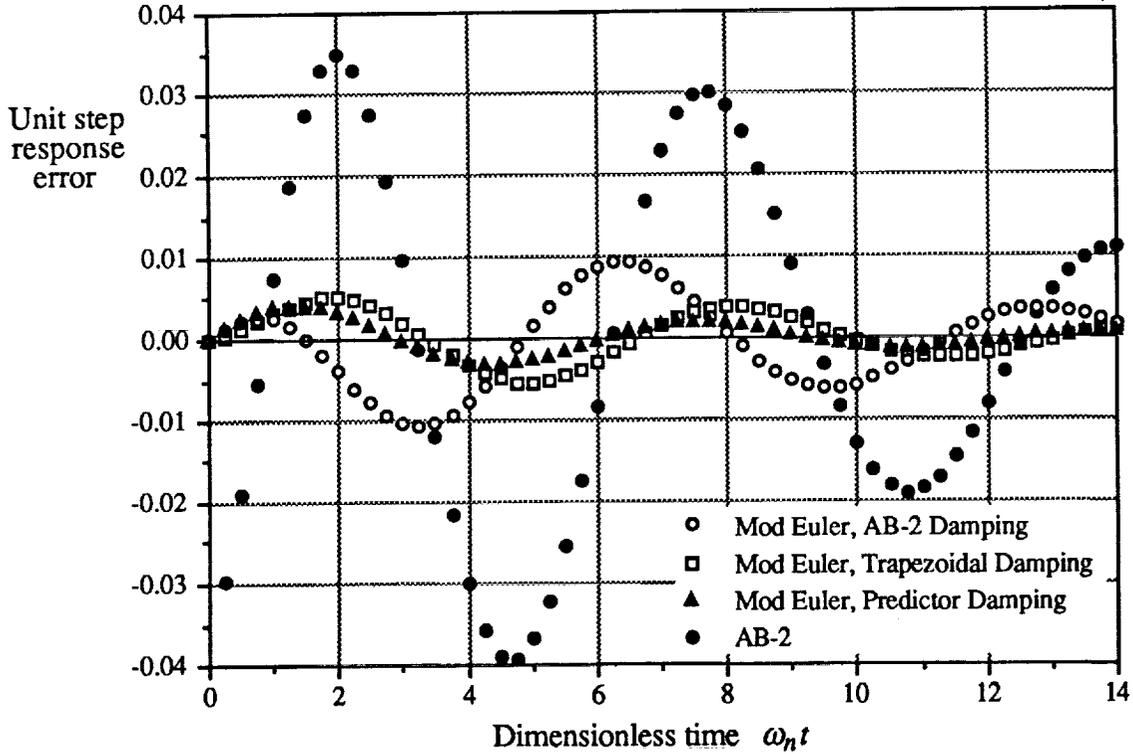


Figure 2. Unit step response errors in simulating second-order system, $\zeta = 0.25$, $\omega_n h = 0.25$.

and the displacement state equations by

$$\dot{\Theta} = Q, \quad \dot{H} = V_p \sin(\Theta - \alpha), \quad \dot{X} = V_p \cos(\Theta - \alpha) \quad (18)$$

Here F_{wx} and F_{wz} are the external force components along the x and z flight-path axes, respectively, and M is the moment about the y body axis; m and I_{yy} represent, respectively, the aircraft mass and pitch-axis moment of inertia. The following formulas were used to represent the external forces and moment:

$$F_{wx} = -qS(C_{D_0} + C_{D_{cL}} C_L^2) - g \sin(\Theta - \alpha) + \frac{T}{m} \cos \alpha \quad (19)$$

$$F_{wz} = qS(C_L + C_{L_{\delta_e}} \delta_e) + g \cos(\Theta - \alpha) - \frac{T}{m} \sin \alpha \quad (20)$$

$$M = qcS(C_{M_0} + C_{M_\alpha} \alpha + C_{M_Q} \frac{c}{2V_p} Q + C_{M_{\dot{\alpha}}} \frac{c}{2V_p} \dot{\alpha} + C_{M_{\delta_e}} \delta_e) \quad (21)$$

where

$$q = \text{dynamic pressure} = \frac{1}{2} \rho V_p^2 \quad (22)$$

and

$$C_L = \text{lift coefficient} = C_{L_0} + C_{L_\alpha} \alpha \quad (23)$$

In these equations S is the aircraft wing area, g is the gravity acceleration, T is powerplant thrust, δ_e is elevator displacement, and c is the mean aerodynamic chord. The various C 's represent aerodynamic coefficients and stability derivatives in accordance with the subscripts. In a full flight-envelope simulation these will be nonlinear functions of other variables such as V_p (through Mach number dependence), α , δ_e , and h .

Based on the way in which modified-Euler integration was introduced in Section 2, the velocity states V_p , α and Q in Eq. (17) would be represented at half-integer frames, with the position states Θ , H and X represented at integer frames. For the n th integration frame this results in the computation of the $n+1/2$ velocity state from the $n-1/2$ velocity state, followed by computation of the $n+1$ position state from the n position state using the $n+1/2$ velocity state just obtained. However, from Eq. (17) it is apparent that it would be better to represent the angle of attack α at integer frames, even though it is derived from a velocity state equation. This is because the dominant term on the right side of Eq. (17) affecting the high-speed dynamics is the pitch-rate Q , which is represented at half-integer frames. The other term in Eq. (24), F_{wz}/mV_p , is the negative of the flight-path-axis pitch rate, and is generally much smaller in magnitude than Q . For this reason we have chosen to represent α at integer frames in the modified Euler mechanization of the flight equations. Since the force (and hence acceleration) term F_{wz} is computed and therefore represented at integer frames, it is necessary to compute an estimate of F_{wz} at the $n+1/2$ frame in the modified Euler integration of α_n to obtain α_{n+1} . This is easily accomplished using the first-order extrapolation formula $F_{wz_{n+1/2}} = (3/2)F_{wz_n} - (1/2)F_{wz_{n-1}}$. The actual difference equations used to solve (17) through (23) with modified Euler integration are presented in a previous paper by the author [4].

As a specific example we consider a business jet flying at 40,000 feet at a speed of Mach 0.7 [5]. For the above flight condition the undamped natural frequency of the short-period mode is about 3 rad/sec and the damping ratio is 0.4. In the example simulation we let the step size $h = 0.1$ second. This makes $\omega_n h \cong 0.3$ for the short-period motion, which should yield the moderate accuracies normally associated with a real-time simulation. We consider the aircraft response to the input function shown in Figure 3, which is a step elevator displacement with a one second rise time. Use of this input function tends to reduce the large transient errors caused by step inputs when predictor integration algorithms are used. It is also probably more typical of an actual transient input. The simulation is started at $t = 0$ with the aircraft in level equilibrium flight. In order to make the example more representative of an ongoing simulation, the step input

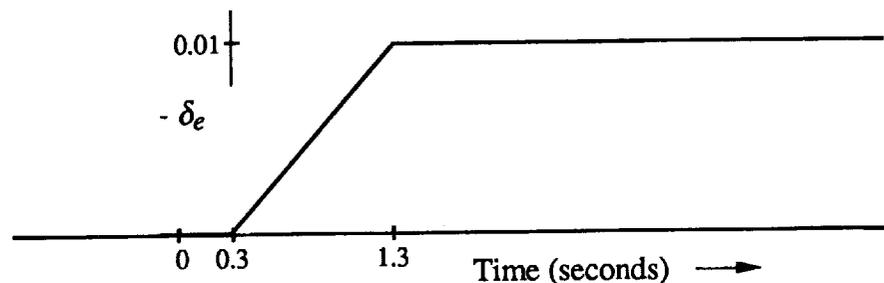


Figure 3. Delayed, finite rise-time step input.

is delayed for 0.3 seconds (three integration steps for $h = 0.1$) after the initial time $t = 0$. Figure 4 shows the error in pitch angle versus time for AB-2 integration and for modified Euler integration using the predictor integration of Table 1 for the integer velocity estimate. We note that the modified Euler method is an order of magnitude more accurate.

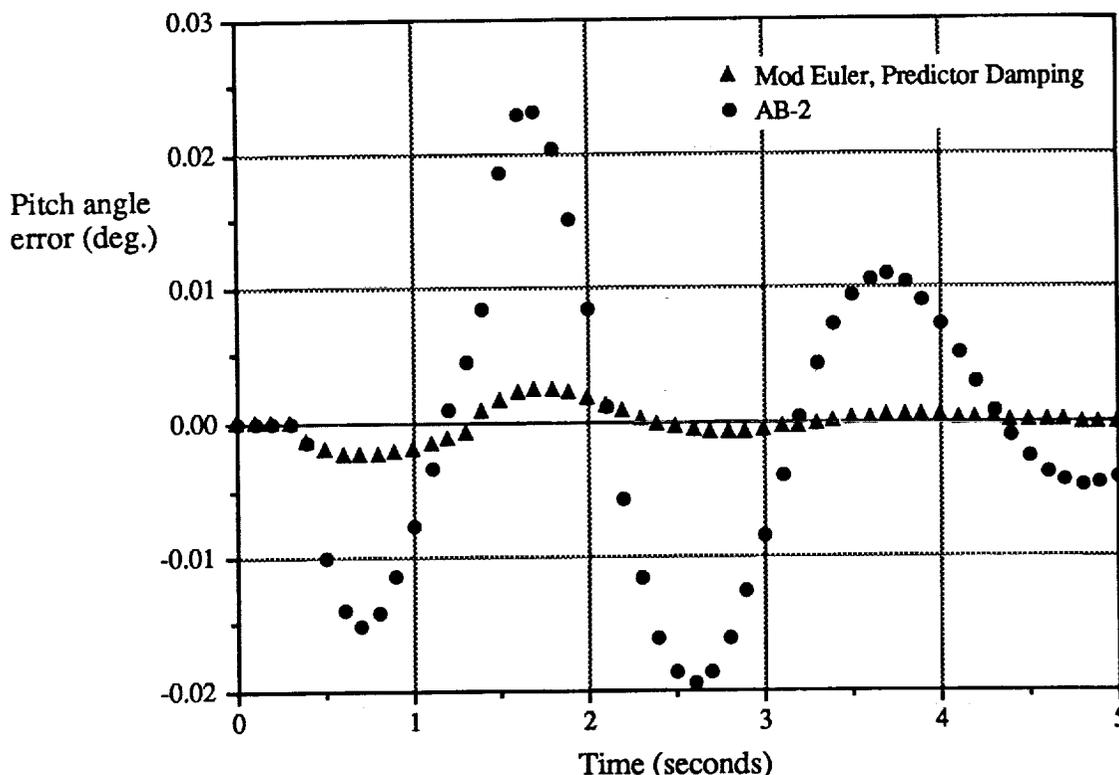


Figure 4. Aircraft pitch angle error for the input function of Figure 3; $h = 0.1$ seconds.

5. Stability Considerations

In addition to considering the dynamic accuracy associated with different numerical integration methods, it is important to consider the stability of the methods. This is usually done by considering the stability boundary in the complex λh plane. These boundaries are shown in Figure 5 for modified Euler integration used to solve Eq. (14) with the various methods for computing the velocity estimate, \hat{v}_n , as presented in Table 1. Also shown in Figure 5 is the stability boundary for the AB-2 predictor method, as well as that for the AM-2 predictor-corrector method. In the latter case the stability region has been reduced by a factor of two to take into account that AM-2 is a two-pass method. Any values of λh lying outside the boundary shown for a given method (the boundaries are symmetric with respect to the real axis) will lead to instability. From the figure it is evident that the modified Euler method with trapezoidal integration for the damping term exhibits the largest stability boundary. Note also that the

stability boundary for all of the modified Euler methods lies on the imaginary axis. This means that modified Euler integration, when used to simulate systems with pure imaginary roots, as in the case of undamped oscillatory modes, will also exhibit pure imaginary roots corresponding to zero damping. This is true regardless of the integration step size h and is the reason why the modified Euler method is especially effective in simulating lightly-damped dynamic systems.

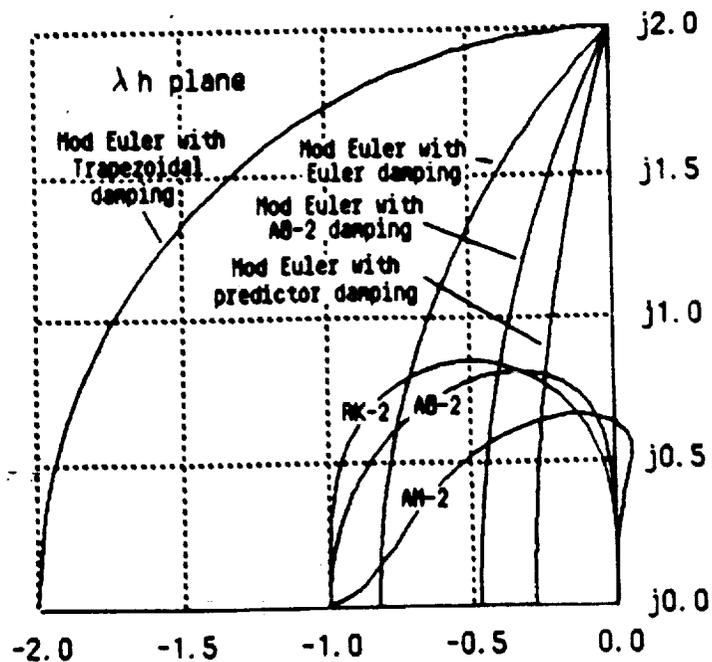


Figure 5. Stability boundaries for modified Euler and other second-order integration methods.

In the second-order system example considered in Section 4 we were able to use trapezoidal integration for the damping term in the modified Euler mechanization because the damping was linear. This in turn permitted us to construct an explicit, single-step formulation represented by Eqs. (15) and (16). When the the dependence of acceleration on velocity is nonlinear, this is no longer possible. Yet it would be advantageous for stability reasons to still use a trapezoidal implementation.

The nonlinear dependence of the acceleration A on the velocity V in Eq. (3) can often be expressed in terms of $V\partial A/\partial V$, where $\partial A/\partial V$ is not a function of V , or at worst is only slightly dependent on V . For example if A represents dQ/dt , the time derivative of pitch rate Q in the flight equations, then $\partial A/\partial Q$ is proportional to the aerodynamic stability derivative C_{MQ} , i.e., the dimensionless pitching moment due to dimensionless pitch rate. C_{MQ} is normally independent of Q , although it may be dependent on other variables such as Mach number. Also, the overall $\partial A/\partial Q$ in this case will be independent of Q . Letting V be a scalar which represents the angular velocity Q , we can rewrite Eq. (14) as follows:

ORIGINAL PAGE IS
OF POOR QUALITY

$$\dot{V} = C_0 [D, U(t)] + C_1 [D, U(t)]V \quad (24)$$

where $C_0 + C_1 V = A$ and $C_1 = \partial A / \partial V$. Now, when mechanizing the modified-Euler difference equations (4) we can compute \hat{V}_n , the estimate of V at the n th frame, by the formula

$$\hat{V}_n = \frac{1}{2} (V_{n+1/2} + V_{n-1/2}) \quad (25)$$

From Eqs. (24) and (25) the difference equation for $V_{n+1/2}$ in Eq. (4) then becomes

$$V_{n+1/2} = V_{n-1/2} + h [C_0 (D_n, U_n) + C_1 (D_n, U_n) \frac{V_{n+1/2} + V_{n-1/2}}{2}] \quad (26)$$

With respect to the velocity state V this equation clearly represents implicit trapezoidal integration. However it can be solved to obtain the following explicit formula for $V_{n+1/2}$:

$$V_{n+1/2} = \frac{(1 + hC_1/2)V_{n-1/2} + hC_0}{1 - hC_1/2} \quad (27)$$

This formulation, i.e., the use of trapezoidal integration for the damping term, expands very substantially the stability region in the λh plane compared with the use of the predictor formula for \hat{V}_n , as we have seen in Figure 5. It can also reduce appreciably the dynamic errors following transient inputs. The extra required computation is modest and consists mainly of an additional division.

In deriving Eq. (27) we have assumed that V is a scalar, whereas V will in general be a vector. In this case $\partial A / \partial V$ will be a matrix, which must be inverted to obtain the explicit formula for $V_{n+1/2}$. Fortunately, the critical terms in this matrix in the case of the flight equations are the diagonal terms, in which case simple formulas similar to Eq. (27) involving only the diagonal terms can be derived. In the longitudinal flight equations (17) through (23), for example, an equation similar to (27) can be written for $Q_{n+1/2}$, where C_1 is proportional to the stability derivative C_{M_Q} .

6. Conclusions

We have shown that mechanical dynamic systems are well suited to a modified Euler integration method which computes displacement and acceleration variables at integer frame times and velocity variables at half-integer frame times. Examination of asymptotic formulas for characteristic root and transfer function errors associated with a linearized version of any nonlinear mechanics problem shows that the modified Euler method is at least twice as accurate as any other known second-order algorithm. For the usual case where the acceleration is a function of velocity, there are a number of candidate methods for computing the required velocity estimates at integer frames from the velocity as computed at half-integer frames. A second-order predictor integration formula produces the most accurate integer-frame velocity estimate; an estimate based on the equivalent of trapezoidal integration produces the most stable simulation. Neither estimate requires any additional derivative evaluations, and the predictor formula can be

used to produce output displacements at half-integer as well as integer frame times in a real-time simulation, i.e., at double the integration frame rate. The modified Euler method is particularly effective in simulating systems with lightly damped modes, since modes with zero damping in a continuous system generate modes with zero damping in the modified Euler mechanization, regardless of the integration step size. The modified Euler method also has a simple and accurate startup procedure and is completely compatible with real-time inputs. Two examples, a second-order linear system and a sixth-order nonlinear flight simulation, have been used to demonstrate the superior accuracy of the modified Euler method.

7. References

1. Gear, William G., "*Numerical Initial Value Problems in Ordinary Differential Equations*," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.
2. Howe, R.M., "Transfer Function and Characteristic Root Errors for Fixed-Step Integration Algorithms," *Transactions of the Society for Computer Simulation*, 2 (4): 293-320, 1985.
3. Fogarty, L.E., and R.M. Howe, "Computer Mechanization of Six-Degree-of-Freedom Flight Equations," *Simulation*, 11(4): 187-193, 1968.
4. Howe, R.M., "An Improved Numerical Integration Method for Flight Simulation," *Proceedings of the AIAA Flight Simulation Technologies Conference*, Boston, Aug. 14-16, 1989, pp 310-316.
5. Roskam, J., "*Airplane Flight Dynamics and Automatic Flight Controls*," Vol. 1, 1982, pp. 616-624, Roskam Aviation and Engineering Corporation, Route 4, Box 274, Ottawa, Kansas 66067.