

A new family of real-time predictor-corrector integration algorithms

R.M. Howe

The University of Michigan
Ann Arbor, Michigan
Applied Dynamics International
Ann Arbor, Michigan

This paper describes a variation of the conventional two-pass explicit Adams-Moulton predictor-corrector integration methods which is suitable for real-time simulation. In this new method the first pass through the state equations uses an Adams-Bashforth type of predictor algorithm to compute an estimate of the state at the $n+1/2$ frame instead of the $n+1$ frame, as is customary. This estimate is then used to compute the derivative at the $n+1/2$ frame which, along with derivatives at the $n, n-1, n-2, \dots$ frames is used in the final corrector pass to calculate the state at the $n+1$ frame. Unlike conventional two-pass Adams-Moulton methods, these new versions are compatible with real-time inputs. The paper shows that they are also superior based on dynamic accuracy measures and stability measures. A three-pass predictor-corrector integration algorithm compatible with real-time inputs is also presented and shown to yield significantly more accurate results than 3rd-order RK (Runge-Kutta) integration.

Introduction

One of the methods sometimes considered as a candidate for numerical integration of ordinary differential equations is the two-pass explicit Adams-Moulton algorithm. In this method the first pass through the state equations in the n th integration frame utilizes an Adams-Bashforth predictor formula to compute an estimate of the state at the $n+1$ frame. In the second pass this estimate is used for the $n+1$ derivative in the implicit Adams-Moulton formula to compute the corrected $n+1$ state. Two-pass explicit Adams-Moulton algorithms of this type exhibit asymptotic accuracy equal to that of implicit Adams-Moulton methods, which can be significantly better than the asymptotic accuracy of Adams-Bashforth corrector methods of the same order [Howe 1985]. Yet the AB (Adams-Bashforth) methods are more popular for real-time digital simulation than the AM (Adams-Moulton) methods. This is primarily because the AB methods are compatible with real-time inputs, whereas the AM methods are not. In this paper we introduce a variation of the two-pass predictor-corrector method which is compatible with real time inputs. It also turns out to be more accurate than the traditional AM methods and has a slightly larger stability region in the λh plane, where λ is an eigenvalue of the system being simulated and h is the step size used in the numerical integration.

In the next section we review the standard two-pass AM predictor-corrector algorithms and present the asymptotic formulas for the dynamic errors associated with the algorithms. In the following sections we present the new real-time predictor-corrector algorithms and show from asymptotic error formulas that they offer improved dynamic accuracy in addition to real-time compatibility. The λh -plane stability boundaries for both standard and real-time predictor-corrector methods are also compared. Finally, a three-

pass predictor-corrector integration algorithm compatible with real-time inputs is presented and shown to yield more accurate results than 3rd-order RK (Runge-Kutta) integration.

Two-pass Explicit Adams-Moulton Algorithms

Assume that the dynamic system being simulated is represented by the following differential equation:

$$\dot{X} = \frac{dX}{dt} = F[X, U(t)] \quad (1)$$

Here X is the state vector and $U(t)$ is the explicit input vector. The second-order two-pass AM predictor-corrector algorithm (hereafter designated as AM-2) employs the following formulas:

$$F_n = F(X_n, U_n), \quad \widehat{X}_{n+1} = X_n + h \left(\frac{3}{2} F_n - \frac{1}{2} F_{n-1} \right)$$

$$\widehat{F}_{n+1} = F(\widehat{X}_{n+1}, U_{n+1}), \quad X_{n+1} = X_n + \frac{1}{2} h (\widehat{F}_{n+1} + F_n) \quad (2)$$

Here $X_n = X(nh)$ and $U_n = U(nh)$, where the integer n represents the integration step or frame index and h is the integration step size. In the first pass through the state equations the derivative F_n is computed using X_n and U_n . This is followed by the calculation of \widehat{X}_{n+1} , which is the estimate for X_{n+1} based on the AB-2 predictor formula. In the second pass \widehat{F}_{n+1} is computed using \widehat{X}_{n+1} and U_{n+1} . This is followed by the corrector calculation of X_{n+1} using trapezoidal integration. Note that $(h/2)(3F_n - F_{n-1})$ in the formula for \widehat{X}_{n+1} represents the area under a linear extrapolation from $t = nh$ to $(n+1)h$ based on F_n and F_{n-1} ; $(h/2)(\widehat{F}_{n+1} + F_n)$ in the formula for X_{n+1} represents the area under the trapezoid based on a linear interpolation between F_n and F_{n+1} . It is straightforward to show that dynamic errors associated with the AM-2 integration algorithm are proportional to $-h^2/12$ [Howe 1985].

For AM-3, the third-order predictor-corrector algorithm, the predictor and corrector passes are based on quadratic extrapolation and interpolation, respectively. This leads to the following formulas:

$$\widehat{X}_{n+1} = X_n + \frac{h}{12} (23F_n - 16F_{n-1} + 5F_{n-2})$$

$$X_{n+1} = X_n + \frac{h}{12} (5\widehat{F}_{n+1} + 8F_n - F_{n-1}) \quad (3)$$

Here F_n and \widehat{F}_{n+1} are defined as in Eq. (2). The dynamic errors for AM-3 integration are proportional to $-h^2/24$.

For AM-4 integration the predictor and corrector passes are based on cubic extrapolation and interpolation, respectively. This leads to the formulas

$$\widehat{X}_{n+1} = X_n + \frac{h}{24} (55F_n - 59F_{n-1} + 37F_{n-2} - 9F_{n-3})$$

$$X_{n+1} = X_n + \frac{h}{24} (9\widehat{F}_{n+1} + 19F_n - 5F_{n-1} + F_{n-2}) \quad (4)$$

Again F_n and \widehat{F}_{n+1} are defined as in Eq. (2). The dynamic errors for AM-4 integration are proportional to $-19h^4/720$.

Dynamic Error Measures for Integration Algorithms

In this section we introduce methods for determining the dynamic errors associated with integration algorithms in general. We begin by considering the solution of the state equation $dx/dt = f(t)$ using a numerical integration formula for x_{n+1} in terms of x_n and the derivative f . We let $x[(n+1)h]$ and $x[nh]$ represent the exact solution of the continuous system at the times $t = (n+1)h$ and nh , respectively. From the appropriate Taylor series expansions we obtain a formula for $x_{n+1} - x_n$ which takes the following form [Gear 1971]:

$$x_{n+1} - x_n \cong x[(n+1)h] - x[nh] - e_1 f_n^{(k)} h^{k+1} \quad (5)$$

Here the term $-e_1 f_n^{(k)} h^{k+1}$ 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 $t = nh$. For example, $k = 1$ and $e_1 = 1/2$ for Euler integration; for trapezoidal integration $k = 2$ and $e_1 = -1/12$. To develop formulas for the global truncation error we use the method of Z transforms [Gilbert 1966]. Taking the Z transform of Eq. (5) and dividing by $z-1$, we obtain

$$X^*(z) \cong X_{ef}^*(z) - \frac{e_1 h^{k+1} F^{(k)}(z)}{z-1} \quad (6)$$

Here $X_{ef}^*(z)$ is the Z transform of the exact solution, $x[nh]$. 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

$$\frac{X^*(e^{j\omega h})}{F^*(e^{j\omega h})} \cong \frac{X_{ef}^*(e^{j\omega h})}{F^*(e^{j\omega h})} - \frac{e_1 h(j\omega h)^k}{e^{j\omega h} - 1} \quad (7)$$

The term X^*/F^* is simply the sinusoidal transfer function, $H_1^*(e^{j\omega h})$, of the numerical integrator. The term $X_{ef}^*/F^* = 1/j\omega$, the transfer function of an ideal integrator. If we now approximate $e^{j\omega h} - 1$ by $j\omega h$, Eq. (8) becomes the following:

$$H_1^*(e^{j\omega h}) \cong \frac{1 - e_1 (j\omega h)^k}{j\omega} \cong \frac{1}{j\omega(1 + e(j\omega h)^k)}, \quad j\omega h \ll 1 \quad (8)$$

Eq. (8) represents a transfer function model which can be used for each numerical integrator. Thus $j\omega$ associated with the ideal integrator is replaced by $j\omega(1 + E)$ for the numerical integrator, where the integrator error E is given by

$$E = e_1 (j\omega h)^k \quad (9)$$

Again, e_1 is the integrator error coefficient and k is the algorithm order. To illustrate the application of this integra-

for transfer function model, we consider numerical simulation of the simple first-order linear system with the differential equation

$$\dot{x} = \lambda x + u(t) \quad (10)$$

with transfer function $H(s)$ given by

$$H(s) = \frac{X(s)}{U(s)} = \frac{1}{s - \lambda} \quad (11)$$

The transfer function $H(j\omega)$ for sinusoidal inputs for the continuous system is

$$H(j\omega) = \frac{1}{j\omega + \lambda} \quad (12)$$

We obtain the equivalent transfer function when using numerical integration by replacing $j\omega$ with $j\omega[1 + e_1(j\omega h)^k]$. Thus the digital transfer function is given approximately by

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

Replacing $j\omega$ in Eq. (13) with λ^* , we obtain simply

$$H^*(e^{\lambda^* h}) \cong \frac{1}{\lambda^*[1 + e_1(\lambda^* h)^k] - \lambda} \quad (14)$$

Here the values of λ^* which make the denominator of $H^*(e^{\lambda^* h})$ vanish are the equivalent characteristic roots of the digital system. Thus we have $\lambda^* \cong \lambda[1 - e_1(\lambda^* h)^k]$. Noting that $\lambda^* \cong \lambda$ when $|\lambda h| \ll 1$, we can write the following approximate formula for e_λ the fraction error in the characteristic root λ :

$$e_\lambda = \frac{\lambda^* - \lambda}{\lambda} \cong -e_1(\lambda h)^k, \quad |\lambda h| \ll 1 \quad (15)$$

Eq. (15) holds for complex as well as real λ and therefore represents a general formula for the characteristic root errors in simulation of any order linear system using an integration algorithm of order k with a known error coefficient e_1 . Note that the formula derivation is based on the replacement of $j\omega$ with $j\omega[1 + e_1(j\omega h)^k]$ in the linear system transfer function $H(j\omega)$ to obtain the digital transfer function $H^*(e^{j\omega h})$. This procedure is valid only for single-pass integration algorithms or for multiple-pass algorithms where the derivatives used in the final pass are based on state-variable estimates obtained with integration algorithms of the same order k as the multi-pass algorithm. This is because the local truncation error associated with intermediate state estimates is of order $k+1$, which in turn means that Eq. (8) is a valid model for integration of both the state X as well as the input U . Thus the substitution procedure works for the single-pass AB predictor and two-pass AM predictor-corrector methods considered here. It does not work for RK methods, where lower-order methods, i.e., Euler integration, are used for intermediate

state estimates.

We next consider the derivation of approximate formulas for transfer function gain and phase errors. From Eq. (13) we can write

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

or

$$\frac{H^*(e^{j\omega h})}{H(j\omega)} - 1 \cong -e_1 \frac{j\omega}{j\omega - \lambda} (j\omega h)^k, \quad \omega h \ll 1 \quad (16)$$

Here $H^*/H - 1$ is a complex number that represents the fractional error in digital transfer function. For $\omega h \ll 1$ it will have a magnitude which is small compared with unity. In this case it is easy to show that the real part of $H^*/H - 1$ is approximately equal to the fractional error in transfer function gain, which we denote as e_M , and the imaginary part is approximately equal to the phase error, which we denote as e_A [Howe 1985]. To obtain $H^*/H - 1$ in the form $e_M + je_A$ we must rationalize the right side of Eq. (16), which means that the general form of the equations for e_M and e_A will depend on whether k is even or odd.

Note that the transfer function for any order linear system can be represented as the product of factors $1/(s + \lambda_p)$ as in Eq. (11) that result from the transfer-function poles, λ_p , multiplied by factors of the form $(1 + \lambda_z)$ that result from the transfer function zeros, λ_z . We have noted above that Eq. (16) can be used to calculate the gain and phase errors associated with each of the first-order pole factors. For zero factors corresponding to the reciprocal of Eq. (11) the gain and phase errors are simply the negative of those given by the real and imaginary parts of $H^*/H - 1$ in Eq. (16), respectively. It follows that the overall digital transfer function gain and phase errors are just the sum of the gain and phase errors, respectively, associated with the individual pole and zero factors of $H(s)$. Thus Eq. (16) can be used to determine quite easily the transfer function gain and phase errors as a function of input frequency ω and step size h in digital simulation of any linear system. We only need to know the order k and error coefficient e_1 associated with the integration algorithm used for the simulation.

In this section, then, we have shown how Eq. (15) can be used to determine characteristic root errors and Eq. (16) to determine transfer function gain and phase errors when simulating linear systems. The former is directly related to transient dynamic errors and the latter to dynamic response errors. Since nonlinear systems in general can often be approximated by linear systems using perturbations with respect to a reference or steady-state solution, the formulas presented here can be used to estimate the dynamic accuracy in simulating nonlinear systems. In particular, they can be used to compare the relative accuracy of different integration methods as reflected by the integrator error coefficient e_1 and the order k associated with each method.

The Real-time Predictor-corrector Algorithms

The AM predictor-corrector algorithms represented by Eqs. (2), (3) and (4) are all two-pass methods, i.e. they require two

(2), (3) and (4) are all two-pass methods, i.e. they require two evaluations of the state-variable derivative F for each integration step. The evaluation of F_n in the first pass requires the input U_n . If U is an external input in a real-time simulation, then U_n can be available at the beginning of the n th integration step. However, the evaluation of F_{n+1} in the second pass requires U_{n+1} , which will not yet be available in a real time simulation when U is an external real-time input. This is because computation of the output state X_{n+1} must be completed prior to $t = (n+1)h$. This in turn means that U_{n+1} as needed for the second pass must be available well in advance of $t = (n+1)h$, usually about one-half frame ahead of time. Thus the AM algorithms are not compatible with real-time inputs in a real-time simulation.

The way around this problem in predictor-corrector methods is to use the predictor pass to compute an estimate of $X_{n+1/2}$ rather than X_{n+1} . The estimate $\widehat{X}_{n+1/2}$ is then used to compute the derivative estimate $\widehat{F}_{n+1/2}$. This derivative also requires the input $U_{n+1/2}$, which now can be available in real time, since it is needed halfway through the integration step.

X_{n+1} is then computed using a corrector pass based on $\widehat{F}_{n+1/2}$, F_n , F_{n-1} , etc. For the real-time predictor-corrector algorithm of order 2, hereafter referred to as RTAM-2, the formulas become the following:

$$F_n = F(X_n, U_n), \quad \widehat{X}_{n+1/2} = X_n + \frac{h}{8}(5F_n - F_{n-1})$$

$$\widehat{F}_{n+1/2} = F(\widehat{X}_{n+1/2}, U_{n+1/2}), \quad X_{n+1} = X_n + h\widehat{F}_{n+1/2} \quad [17]$$

Here the term $(h/8)(5F_n - F_{n-1})$ in the predictor formula for $\widehat{X}_{n+1/2}$ represents the area under a linear extrapolation from $t = nh$ to $(n+1/2)h$ based on F_n and F_{n-1} . The corrector pass for computing X_{n+1} uses the derivative estimate $\widehat{F}_{n+1/2}$ halfway through the frame in a modified Euler or rectangular formula [Smith 1977, Howe 1988]. It is interesting to note that the dynamic errors associated with this RTAM-2 integration are proportional to $h^2/24$ compared with $-h^2/12$ for standard AM-2 integration, i.e. twice as accurate. This results from an integrator error coefficient given by $e_1 = 1/24$ for RTAM-2 integration.

If Euler integration with a step size of $h/2$ is used for the first pass in Eq. (17), the formulas become the following:

$$F_n = F(X_n, U_n), \quad \widehat{X}_{n+1/2} = X_n + \frac{h}{2}F_n$$

$$\widehat{F}_{n+1/2} = F(\widehat{X}_{n+1/2}, U_{n+1/2}), \quad X_{n+1} = X_n + h\widehat{F}_{n+1/2} \quad [18]$$

Eq. (18) is a version of second-order Runge-Kutta integration sometimes called RTRK-2 because of its compatibility with real-time inputs U . For this integration method the asymptotic formula for the characteristic root error, when compared with Eq. (15), indicates an integrator error coefficient $e_1 = h^2/6$,

compared with $h^2/24$ for RTAM-2. This lower accuracy results from the larger error associated with the use of first-order Euler integration rather than the second-order predictor for the computation of $\widehat{X}_{n+1/2}$.

For RTAM-3 the predictor estimate of $X_{n+1/2}$ is based on the area from $t = nh$ to $t = (n+1/2)h$ under a quadratic approximation for $F(t)$ which passes through F_n , F_{n-1} and F_{n-2} . This is followed by the corrector formula for X_{n+1} , which is based on the area from $t = nh$ to $t = (n+1)h$ under a quadratic approximation for $F(t)$ which passes through $\widehat{F}_{n+1/2}$, F_n and F_{n-1} . This leads to the following equations:

$$\widehat{X}_{n+1/2} = X_n + \frac{h}{24}(17F_n - 7F_{n-1} + 2F_{n-2})$$

$$X_{n+1} = X_n + \frac{h}{18}(20\widehat{F}_{n+1/2} - 3F_n + F_{n-1}) \quad [19]$$

Here F_n and $\widehat{F}_{n+1/2}$ are defined as in Eq. (17). For RTAM-3 the integrator error coefficient $e_1 = h^3/36$, compared with $-h^3/24$ for conventional AM-3.

In the case of RTAM-4 the predictor estimate of $X_{n+1/2}$ is based on the area from $t = nh$ to $t = (n+1/2)h$ under a cubic approximation for $F(t)$ which passes through F_n , F_{n-1} , F_{n-2} and F_{n-3} . This is followed by the corrector formula for X_{n+1} , which is based on the area from $t = nh$ to $t = (n+1)h$ under a cubic

approximation for $F(t)$ which passes through $\widehat{F}_{n+1/2}$, F_n , F_{n-1} and F_{n-2} . This then leads to the following equations:

$$\widehat{X}_{n+1/2} = X_n + \frac{h}{384}(297F_n - 187F_{n-1} + 107F_{n-2} - 25F_{n-3})$$

$$X_{n+1} = X_n + \frac{h}{30}(36\widehat{F}_{n+1/2} - 10F_n + 5F_{n-1} - F_{n-2}) \quad [20]$$

Once again F_n and $\widehat{F}_{n+1/2}$ are defined as in Eq. (17). For RTAM-4 the integrator error coefficient $e_1 = 59h^4/2880$, compared with $-19h^4/720$, i.e. $-76h^4/2880$, for conventional AM-4.

From the above results it is apparent that the RTAM predictor-corrector integration algorithms introduced in this section, in addition to being compatible with real-time inputs, are more accurate than the conventional AM predictor-corrector methods. In particular we have seen that the ratio of RTAM to AM integrator error coefficients is $1/2$, $2/3$, and $59/76$ for the 2nd, 3rd and 4th order algorithms, respectively.

Before comparing the stability boundaries in the λh plane for RTAM and AM integration, we consider a real-time three-pass integration method based on RTAM-2 which has a significant accuracy advantage over RK-3 integration.

Real-time Three-pass Predictor-Corrector

The performance of the two-pass RTAM-3 integration method suggests that a similar algorithm be used to provide an estimate of $X_{n+2/3}$ instead of X_{n+1} . Then a third pass based on F_n and $\widehat{F}_{n+2/3}$ can be used to compute X_{n+1} . The

$$\begin{aligned}
F_n &= F(X_n, U_n), \quad \widehat{X}_{n+1/3} = X_n + \frac{h}{324} (137F_n - 40F_{n-1} + 11F_{n-2}) \\
\widehat{F}_{n+1/3} &= F(X_{n+1/3}, U_{n+1/3}), \quad \widehat{X}_{n+2/3} = X_n + \frac{h}{54} (39\widehat{F}_{n+1/3} - 4F_n + F_{n-1}) \\
\widehat{F}_{n+2/3} &= F(\widehat{X}_{n+2/3}, U_{n+2/3}), \quad X_{n+1} = X_n + \frac{h}{4} (F_n + 3\widehat{F}_{n+2/3}) \quad [21]
\end{aligned}$$

Here the first-pass estimate for $\widehat{X}_{n+1/3}$ is based on the area from $t = nh$ to $t = (n+1/3)h$ under a quadratic approximation for $F(t)$ which passes through F_n, F_{n-1} and F_{n-2} . The second-

pass estimate for $\widehat{X}_{n+2/3}$ is based on the area from $t = nh$ to $t = (n+2/3)h$ under a quadratic approximation for $F(t)$ which

passes through $\widehat{F}_{n+1/3}, F_n$ and F_{n-1} . The final pass for X_{n+1} is based on the area from $t = nh$ to $t = (n+1)h$ under a quadratic

approximation for $F(t)$ which passes through $\widehat{F}_{n+2/3}, \widehat{F}_{n+1/3}$ and F_n . For this real-time three-pass predictor-corrector algorithm of Eq. (18) the integrator error coefficient $e_i = h^3/216$ and the algorithm order $k = 3$. Note that the method is compatible with real-time inputs, since U_n is required at the beginning of the first pass, $U_{n+1/3}$ at the beginning of the second pass, and $U_{n+2/3}$ at the beginning of the third pass.

It is useful to compare the above integration method with the version of RK-3 integration given by the following formulas [Howe 1985]:

$$\begin{aligned}
F_n &= F(X_n, U_n), \quad \widehat{X}_{n+1/3} = X_n + \frac{h}{3} F_n \\
\widehat{F}_{n+1/3} &= F(X_{n+1/3}, U_{n+1/3}), \quad \widehat{X}_{n+2/3} = X_n + \frac{2h}{3} \widehat{F}_{n+1/3} \\
\widehat{F}_{n+2/3} &= F(\widehat{X}_{n+2/3}, U_{n+2/3}), \quad X_{n+1} = X_n + \frac{h}{4} (F_n + 3\widehat{F}_{n+2/3}) \quad [22]
\end{aligned}$$

Here $\widehat{X}_{n+2/3}$ is computed in the first two passes using the RTRK-2 algorithm given in Eq. (18), but with a step size of $2h/3$ rather than h . The formula for final pass which computes X_{n+1} is the same as in Eq. (21). For the RK-3 integration method of Eq. (22) the asymptotic formula for the characteristic root error, when compared with Eq. (15), indicates an integrator error coefficient $e_i = h^2/24$, versus $h^2/216$ for the three-pass method given previously by Eq. (21). This lower accuracy results from the larger error associated with the use of second-order RTRK-2 integration rather than the third-

order predictor for the computation of $\widehat{X}_{n+2/3}$.

An interesting variation of the three-pass method of Eq. (21) is to use a second-order rather than third-order

predictor formula for computing $\widehat{X}_{n+1/3}$ in the first pass. In this case the formulas become

$$F_n = F(X_n, U_n), \quad \widehat{X}_{n+1/3} = X_n + \frac{h}{18} (7F_n - F_{n-1})$$

$$\begin{aligned}
\widehat{F}_{n+1/3} &= F(X_{n+1/3}, U_{n+1/3}), \quad \widehat{X}_{n+2/3} = X_n + \frac{h}{54} (39\widehat{F}_{n+1/3} - 4F_n + F_{n-1}) \\
\widehat{F}_{n+2/3} &= F(\widehat{X}_{n+2/3}, U_{n+2/3}), \quad X_{n+1} = X_n + \frac{h}{4} (F_n + 3\widehat{F}_{n+2/3}) \quad [23]
\end{aligned}$$

Here the integrator error coefficient is still given by $e_i = h^3/216$, since a third-order predictor formula is used to compute $\widehat{X}_{n+2/3}$ in the second pass. But now the stability boundary in the λh plane is slightly larger, as we shall see in the next section.

Stability Boundaries for Predictor-Corrector Algorithms

Until now we have concentrated on comparing the dynamic accuracy of various real-time and non real-time predictor-corrector algorithms. Another important consideration is the stability of the algorithms. When the integration step size h becomes too large, instability can result, either from changes in one of the roots corresponding to a continuous system eigenvalue or from one of the extraneous roots introduced by the predictor integration algorithm. For example, in the AM-2 algorithm of Eq. (2) the dependence of

\widehat{X}_{n+1} on F_n as well as F_n introduces an additional discrete state for each state of the continuous system, assuming F is a function of X . The corresponding extraneous characteristic root can cause instability when h becomes too large, even when the main root remains fairly close to its original value. To depict the stability range of numerical integration methods it is customary to plot the stability boundary in the complex λh plane [Benyon 1968]. The calculation of this boundary is most easily accomplished by considering the Z transform of the digital system when solving the simple first-order system of Eq. (10), where in general the complex eigenvalue $\lambda = \lambda_{\text{real}} + j\lambda_{\text{imag}}$. If we apply the AM-2 integration algorithm represented by Eq. (2) to Eq. (10) and take the Z transform of the resulting difference equations, the following formula for $H^*(z)$, the digital system Z transform, is obtained:

$$\frac{X^*(z)}{F^*(z)} = H^*(z) = \frac{\frac{h}{2} \left(z^2 + \left(1 + \frac{3}{2} \lambda h \right) z - \frac{1}{2} \lambda h \right)}{z^2 - \left(1 - \lambda h + \frac{3}{4} \lambda^2 h^2 \right) z + \frac{1}{4} \lambda^2 h^2} \quad [24]$$

The digital system will be unstable when one or more of the poles of $H^*(z)$ lie outside the unit circle in the z plane [Gilbert 1966]. Thus the stability boundary in the λh plane will be defined by the values of λh for which the denominator of $H^*(z)$ vanishes when $|z| = 1$, i.e. when $z = e^{j\theta}$. To obtain these λh values we let $z = e^{j\theta}$ in the denominator of Eq. (24), set the denominator equal to zero, and solve for $\lambda_{\text{real}} h$ and $\lambda_{\text{imag}} h$ for θ values ranging from 0 to 2π . The boundary has been plotted in Figure 1 for AM-2, RTAM-2 and AB-2 integration. Any points lying outside the boundary for each of the algorithms signify instability for that particular λh . The stability boundaries are, of course, symmetric with respect to the real axis.

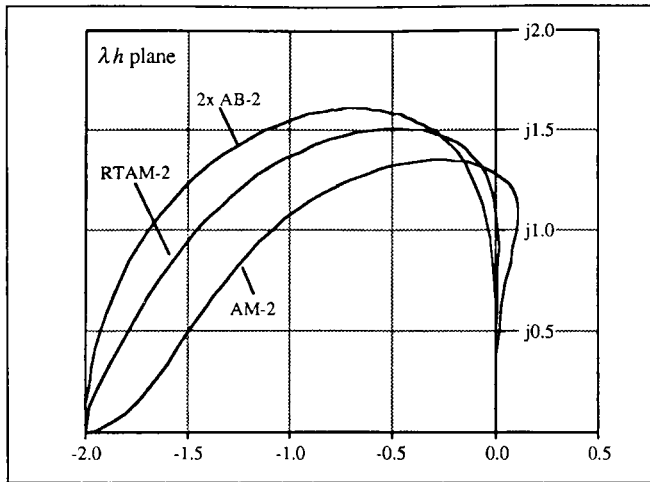


Figure 1. Stability boundaries for AB-2 (normalized), AM-2 and RTAM-2 integration

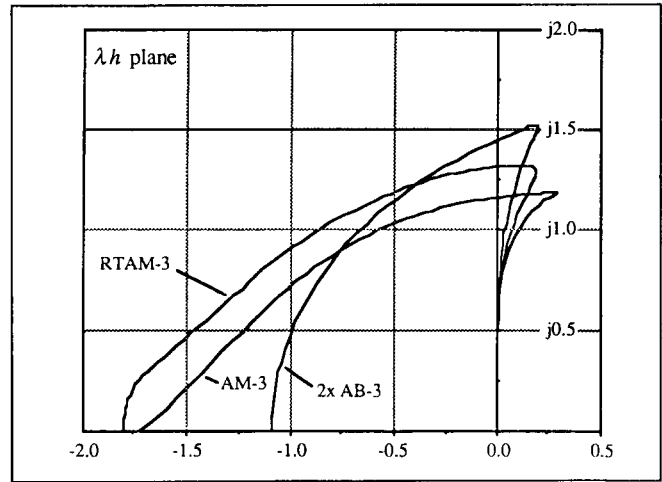


Figure 2. Stability boundaries for AB-3 (normalized), AM-3 and RTAM-3 integration

In the case of AB-2 integration the stability boundary shown in Figure 1 has been doubled in size to reflect the fact that AB-2 integration is a single-pass method, whereas AM-2 and RTAM-2 are two-pass methods. Because of this a given digital processor will be able to execute a real-time AB-2 simulation at twice the frame rate and hence half the mathematical step size in comparison with an AM-2 or RTAM-2 simulation. With this factor of two normalization the AB-2 boundary is somewhat larger than the RTAM-2 boundary, which in turn is larger than the stability boundary for conventional AM-2 integration.

The stability boundaries for AM-3, RTAM-3 and AB-3 are shown in Figure 2. Again the AB-3 boundary has been doubled in size to reflect the fact that the algorithm executes twice as fast as either AM-3 or RTAM-3. In all three of these methods the integration formulas depend on F_{n-1} and F_{n-2} in addition to F_n . Thus in each case there are two additional states and hence two extraneous roots for each state in the continuous system being simulated. Portions of the stability boundaries in Figure 2 are due to these extraneous roots. The stability boundaries have been calculated using the same methodology described above for Figure 1. From Figure 2 it is evident that the RTAM-3 algorithm has the largest stability area, followed by conventional AM-3 and AB-3 integration.

The stability boundaries for AM-4, RTAM-4 and AB-4 are shown in Figure 3. Once again the AB-4 boundary has been doubled in size to reflect the fact that the algorithm executes twice as fast as either AM-4 or RTAM-4. Here all of the methods use integration formulas that depend on F_{n-1} , F_{n-2} and F_{n-3} in addition to F_n . Thus in each case there are three additional states and therefore three extraneous roots for each state in the continuous system being simulated. As in Figure 2, portions of the stability boundaries in Figure 3 are due to these extraneous roots. From Figure 3 it is clear that the RTAM-4 has the largest area of stability, with both RTAM-4 and AM-4 exhibiting substantially larger stability area than AB-4.

Finally, in Figure 4 are shown the stability boundaries for the two real-time three-pass predictor-corrector algorithms introduced in the previous section. Also shown in the figure

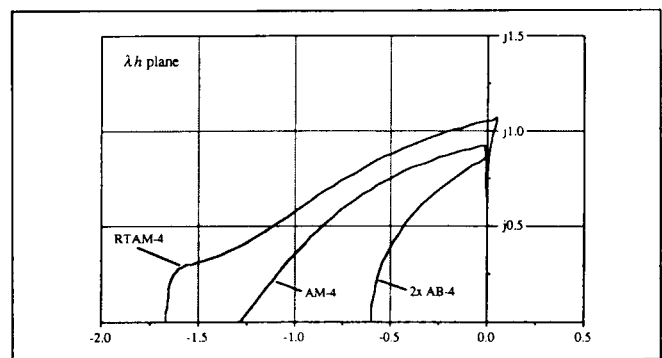


Figure 3. Stability boundaries for AB-4 (normalized), AM-4 and RTAM-4 integration

for comparison purposes are the stability boundaries for RK-3 and AB-3 integration. The AB-3 boundary has been tripled in size to reflect the fact that single-pass AB-3 algorithm executes three times as fast as the other three-pass algorithms. From Figure 4 it is evident that all four algorithms have comparable stability boundaries, with RK-3 having the largest and AB-3 (normalized) the smallest. Thus the two real-time three-pass predictor-corrector algorithms suffer only a slight degradation in stability in return for a factor of nine improvement over RK-3 in eigenvalue accuracy.

Accuracy of the Asymptotic Formulas for Dynamic Errors

The formulas presented in Eqs. (15) and (16) for the dynamic errors associated with integration algorithms are only valid when the integration step size h is sufficiently small ($|\lambda h| \ll 1$ for characteristic root errors, $\omega h \ll 1$ for transfer function errors). For the different integration methods described in the paper, Table 1 summarizes the formulas for the approximate errors in the characteristic roots. In the last column of the table the integrator error coefficient, e_p , has been normalized through multiplication by N^k , where

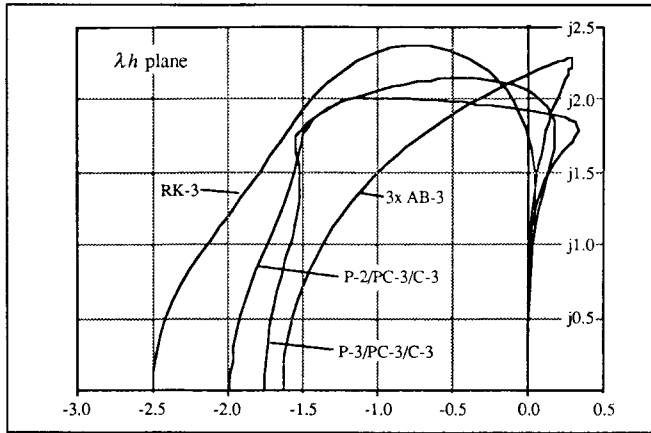


Figure 4. Stability boundaries for AB-3 (normalized), RK-3 and two real-time three-pass predictor corrector algorithms, one with second-order prediction used for the first pass (P-2/PC-3/C-3), the other with third-order prediction used for the first pass (P-3/PC-3/C-3).

N is the number of passes (state-variable derivative calculations) associated with the given integration algorithm. For example, $N = 2$ and $k = 2$ for AM-2 integration, since it is a two-pass algorithm of second order. This means that in general it will require twice as long to execute as the single-pass AB-2 method, which in turn means that the mathematical step size h in a real-time simulation will need to be twice as large. Since the root error is proportional to h^k , i.e., h^2 in this case, the error will be $2^2 = 4$ times larger than it would be for the same e_1 were the algorithm single-pass. Thus among algorithms of the same order the normalized error coefficient is a true indication of the relative accuracy of each method for small step size ($|\lambda h| \ll 1$). Based on asymptotic errors in the characteristic root, then, it is apparent from the last column in Table 1 that RTAM-2 is the best second-order method, either of the three-pass predictor-corrector algorithms represent the best third-order method, and RTAM-4 is the best fourth-order method.

The asymptotic formula for the fractional error in sinusoidal transfer function was derived earlier as Eq. (16). We recall that this formula is valid for the e_1 values listed in Table 1 for all the integration algorithms except RK-2 and RK-3, where a different formula applies (Howe 1985). This is because the RK methods use lower-order integration algorithms for the intermediate-pass integrations, as pointed out earlier in the section on dynamic error measures.

It should be noted that there can be significant differences between various integration methods in regard to the range of $|\lambda h|$ or ωh over which the asymptotic error formulas are valid. This is an especially important consideration in real-time simulation, where the accuracy requirements are often modest and where relatively large integration step sizes are therefore frequently used. The $|\lambda h|$ range of validity of the asymptotic formulas for the fractional error in characteristic root, e_{λ} , is illustrated in Figure 5. The curves shown in the figure for each integration method are based on exact values, λ^* , of digital system roots when solving Eq. (10) numerically for λ negative real. In each case the exact λ^* is computed from the appropriate pole, z_1 , of the digital system Z transform, $H^*(z)$, using the following formula [Gilbert 1966]:

$$\lambda^* = \frac{1}{h} \ln(z_1) \quad [25]$$

Table 1. Summary of Asymptotic Formulas for Characteristic Root Errors

Integration algorithm	Reference for algorithm formula	No. of passes	Algorithm order, k	e_1 (error coefficient)	e_1 (normalized for single pass)
AB-2	Eq. (2), first pass	1	2	$\frac{5}{12}$	0.41667
AM-2	Eq. (2)	2	2	$-\frac{1}{12}$	-0.33333
RTAM-2	Eq. (17)	2	2	$\frac{1}{24}$	0.16667
RTRK-2	Eq. (18)	2	2	$\frac{1}{6}$	0.66667
AB-3	Eq. (3), first pass	1	3	$\frac{3}{8}$	0.37500
AM-3	Eq. (3)	2	3	$-\frac{1}{24}$	-0.33333
RTAM-3	Eq. (19)	2	3	$\frac{1}{36}$	0.22222
P-3/PC-3/C-3	Eq. (21)	3	3	$\frac{1}{216}$	0.12500
P-2/PC-3/C-3	Eq. (22)	3	3	$\frac{1}{216}$	0.12500
RK-3	Eq. (23)	3	3	$\frac{1}{24}$	1.12500
AB-4	Eq. (4), first pass	1	4	$\frac{251}{720}$	0.34861
AM-4	Eq. (4)	2	4	$-\frac{19}{720}$	-0.42222
RTAM-4	Eq. (20)	2	4	$\frac{59}{2880}$	0.32778

For example, in the case of AM-2 integration, $H^*(z)$ is given by Eq. (24). For a specific λh we can solve for the two roots of the denominator of $H^*(z)$. One of these roots corresponds to z_1 , from which we can determine λ^* with Eq. (25). Using Eq. (15), we can then determine the fractional root error e_{λ} , which will be exact. In Figure 5 we have plotted normalized root errors as $N^k e_{\lambda} / |\lambda h|^k$, where N as before is the number of state-equation passes per integration step associated with the given k th-order algorithm. For AM-2 integration, $N = 2$ and $k = 2$. Hence the normalized root error for AM-2 in Figure 5a is plotted as $4e_{\lambda} / |\lambda h|^2$. From Table 1 we see that for AM-2 integration the asymptotic formula for e_{λ} is $-e_1(\lambda h)^2 = (\lambda h)^2 / 12$. Thus $N^k e_{\lambda} / |\lambda h|^k = 4e_{\lambda} / |\lambda h|^2 = 4/12 = 0.333$, which is indeed the value for the normalized AM-2 root error in Figure 5a when $|\lambda h| = 0$. As $|\lambda h|$ increases, the normalized error increases. This is because the asymptotic formula is less accurate for larger $|\lambda h|$ values. N is also equal to 2 for the two-pass RTAM-2 and RTRK-2 methods in Figure 5a, whereas $N = 1$ for the single-pass AB-2 algorithm.

From Figure 5a it is apparent for $|\lambda h| < 0.7$ that the RTAM-2 algorithm exhibits the smallest characteristic-root error. It is considerably more accurate than the conventional AM-2 predictor-corrector method. Only for $|\lambda h| > 0.7$ does the AB-2 algorithm exhibit smaller root errors than RTAM-2. For

these large step sizes, however, the root errors will be very large indeed.

Figure 5b for the third-order algorithms shows that the three-pass predictor-corrector methods (P-2/PC-3/C-3 or P-3/PC-3/C-3) as introduced in this paper perform better than

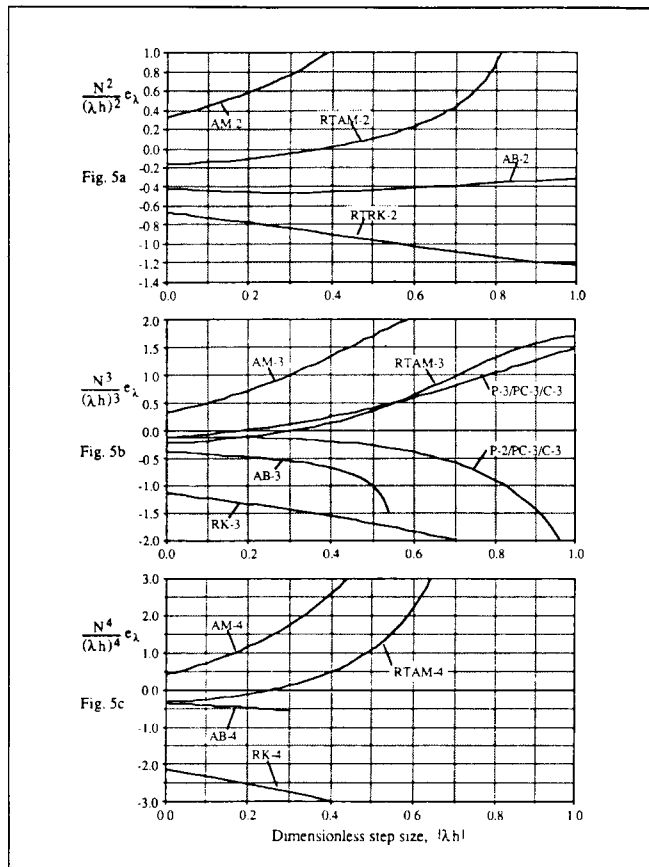


Figure 5. Normalized characteristic root errors for 2nd, 3rd, and 4th-order integration algorithms.

any of the other third-order methods, based on characteristic-root errors. This is true in spite of the penalty associated with three passes ($N = 3$) compared with $N = 2$ for AM-2 and RTAM-2, and $N = 1$ for AB-3. From Figure 5c for fourth-order algorithms we see that the RTAM-4 introduced in this paper exhibits the smallest normalized characteristic root errors.

We consider next the ωh range of validity of the asymptotic formulas for sinusoidal transfer function errors. Exact values for gain and phase errors can be calculated from $H^*(e^{i\omega h})$, where $H^*(z)$ is the digital system Z transform. The approximate formulas for $\omega h \ll 1$ are determined from Eq. (16), except, as noted earlier, in the case of the RK methods. Here a graphical comparison of exact and approximate normalized gain and phase errors is more complicated, since the errors depend on both ωh and λh . In general, however, the trends evident in Figure 5 for characteristic root errors are similar when normalized transfer function gain and phase errors are plotted versus λh for fixed values of ωh .

Comparison of Time-domain Errors

To this point we have only compared the dynamic accuracy of the different integration algorithms by considering characteristic root errors, and sinusoidal transfer function gain and phase errors. It is also tempting to make dynamic accuracy comparisons based on time-domain errors in simulating the response of a given system to a specific time-dependent input. For example, the errors in computed response to a step-function input are sometimes considered. However, unless one is specifically interested in step-function response accuracy, this can be a very misleading method for dynamic accuracy comparison in the case of integration algorithms which employ predictor methods. This is because predictor algorithms of order two and higher will have large transient errors introduced by the step input. The accuracy of predictor methods depends on the existence of input derivatives, a condition which is violated by a discontinuous input. In fact, the dynamic accuracy of the time-domain response is often completely dominated by the startup transient generated as a result of a step input.

To avoid this difficulty we will consider in this section the errors of the various integration methods in simulating the response of a dynamic system to a step displacement input with limited acceleration. For simplicity we let the dynamic system be represented by the following second-order linear system:

$$\dot{X} = Y, \quad \dot{Y} = \omega_n^2 [U(t) - X] - 2\zeta\omega_n Y \quad [26]$$

Here X is the output, $U(t)$ is the input, ω_n is the undamped natural frequency, and ζ is the damping ratio. We let $U(t)$ be the acceleration-limited unit step function shown in Figure 6, where the limit on input acceleration is given by

$$\begin{aligned} \ddot{U} &= \frac{1}{T^2}, 0 \leq t < T, \\ &= -\frac{1}{T^2}, T \leq t < 2T, \\ &= 0, t \geq 2T. \end{aligned} \quad [27]$$

When integrated twice with T set equal to $1.2/\omega_n$, Eq. (27) yields the input shown in Figure 6. The ideal second-order

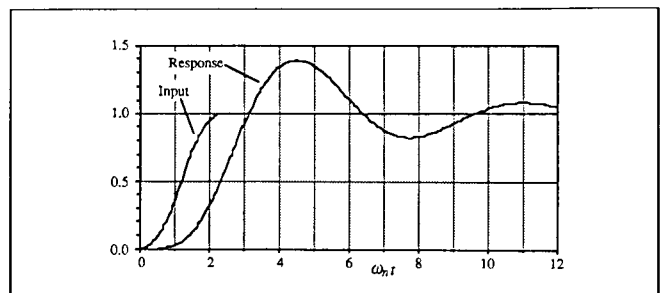


Figure 6. Response of second-order system ($\zeta=0.25$) to an acceleration-limited step input.

system response to this input with initial conditions $X(0) = Y(0) = 0$ is also shown in the figure. The response is very nearly the same as that obtained when a true unit step input is applied at $t = T$, i.e. at $\omega_n t = 1.2$ in Figure 6. Thus the use of this acceleration-limited step input removes input discontinuities in displacement, slope, and second derivative without changing significantly the shape of the second-order system unit step response. It can also be argued that the acceleration-limited step input is a more realistic input for a mechanical dynamic system.

We consider first the performance of second-order integration methods. The response errors for AB-2, AM-2, RTAM-2, and RTRK-2, as given, respectively, by the predictor-pass formula in Eq. (2) and the predictor-corrector-pass formulas in Eqs. (2), (17) and (18) are plotted versus $\omega_n t$ in Figure 7. Initial conditions on X and Y are assumed to be zero for $t \leq 0$. This is representative of a second-order system in stable equilibrium with zero input to which the acceleration-limited step input is applied at $t = 0$. Thus $X_0 = Y_0 = X_{-1} = Y_{-1} = 0$, which solves the second-order predictor startup problem. In Figure 7 the integration step size $\omega_n h = 0.1$ for the single-pass AB-2 method and 0.2 for the two-pass AM-2, RTAM-2 and RTRK-2 methods. As noted earlier, this is because the single-pass AB-2 method will execute approximately twice as fast as the two-pass algorithms on a given processor. The results in Figure 7 show that the RTAM-2 predictor-corrector method introduced in this paper is the most accurate as measured by the response errors for an acceleration-limited step input.

Next we consider the performance of third-order predictor-corrector methods. The response errors for AB-3, AM-3, and

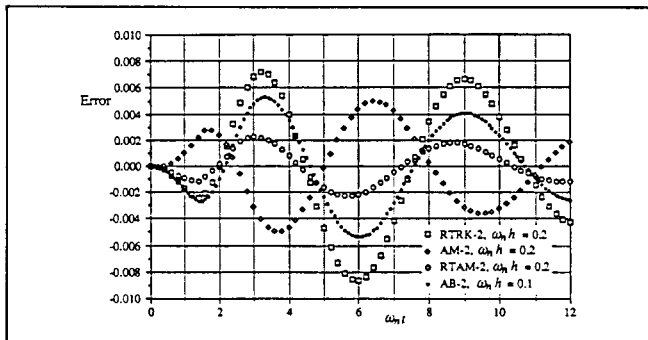


Figure 7. Step-response errors for second-order integration algorithms.

RTAM-3, as given, respectively, by the predictor-pass formula in Eq. (3) and the predictor-corrector-pass formulas in Eqs. (3) and (19) are shown in Figure 8. Again, initial conditions on X and Y are assumed to be zero for $t \leq 0$. Thus $X_0 = Y_0 = X_{-1} = Y_{-1} = X_{-2} = Y_{-2} = 0$ for the third-order predictor startups. As in Figure 7, the integration step size $\omega_n h = 0.1$ for the AB-2 method and 0.2 for the two-pass AM-2 and RTAM-2 methods. Again, Figure 8 shows the superiority of the RTAM-2 algorithm.

In Figure 9 we compare the real-time, three-pass methods given in Eqs. (21) and (23) with RK-3 integration as given in Eq. (22) and AB-3 integration. Here the integration step size $\omega_n h = 0.3$ for the three-pass algorithms and 0.1 for the single-

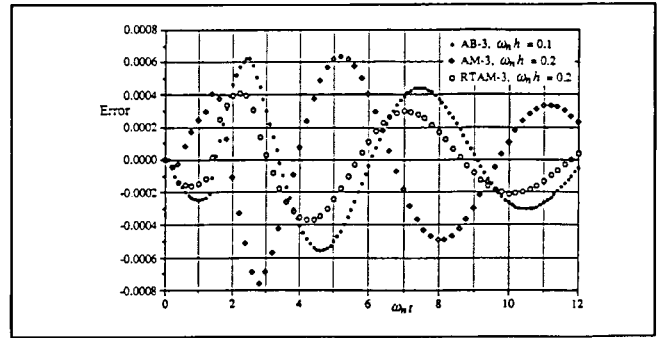


Figure 8. Step-response errors for third-order predictor-corrector integration algorithms.

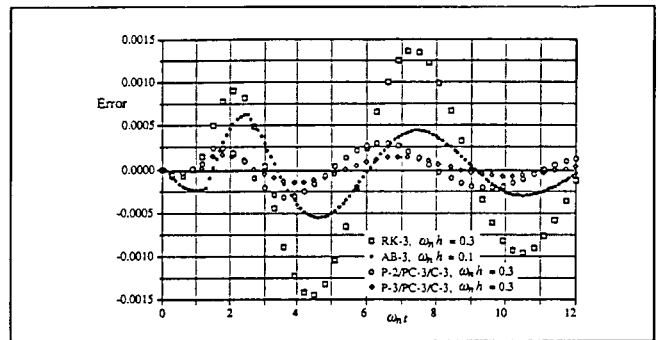


Figure 9. Step-response errors for three-pass third-order integration algorithms.

pass AB-3 algorithm. We note that the P-3/PC-3/C3 method of Eq. (21) gives the most accurate response to the acceleration-limited step input, followed closely by the P-2/PC-3/C3 method of Eq. (23). Both are significantly more accurate than AB-3 despite the fact that their step size is three times larger to reflect actual processor time requirements per integration step. The RK-3 method is the least accurate of all for this example case.

The step-response errors in Figures 7 through 9 in general reflect the relative sizes of the normalized eigenvalue error coefficients e_i shown in the last column of Table 1. An additional advantage to the two and three-pass predictor-corrector algorithms introduced in this paper results from their use of higher order predictor methods for the intermediate integration passes. As a result, the outputs for these intermediate passes can be used to provide accurate estimates of the states at intermediate times during each step. This is illustrated in Figure 10 for the three-pass P-3/PC-3/C3 method. For comparison purposes, similar results are also plotted for the RK-3 method. The figure shows the output errors one-third and two-thirds of the way through each integration step as well as the errors at the integer step times. The much larger scatter in the RK-3 data results from the fact that first-order Euler integration is used in the first pass and second-order RTRK-2 integration is used in the second pass, as can be seen in Eq. (22). Conversely, the P-3/PC-3/C-3 method of Eq. (21) uses third-order methods for all three passes.

Thus the three-pass predictor-corrector methods can be used to provide outputs at three times the basic integration

frame rate. These outputs can in turn be used to drive D to A (digital-to-analog) converters in a real-time simulation requiring continuous outputs. The resulting dynamic errors in the continuous outputs will be much smaller than the errors when only the outputs at integer frame times are used to drive the DAC's.

Howe, R.M. 1988. "Simulation of Linear Systems Using Modified Euler Integration Methods," *Transactions of the Society for Computer Simulation*, Vol. 5, No. 2, April, 1988, pp. 125-152.
 Smith, Jon M. 1977. *Mathematical Modeling and Digital Simulation for Engineers and Scientists*, John Wiley & Sons, New York.

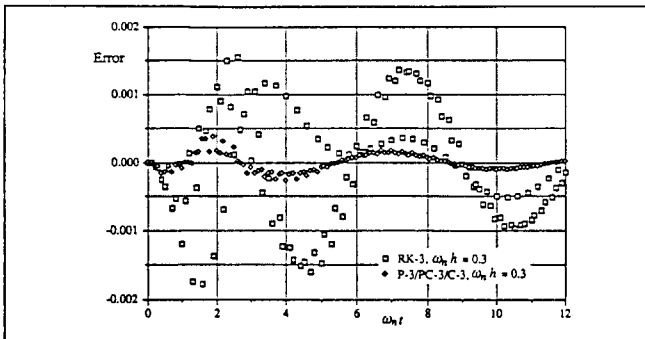


Figure 10. Three-pass P-3/PC-3/C-3 and RK-3 integration data with step response errors shown at each pass.

Conclusion

In this paper we have introduced new two-pass predictor-corrector integration algorithms which, unlike conventional two-pass AM (Adams-Moulton) algorithms, are compatible with external inputs in a real-time simulation. In each case, starting with the state at frame n , the new algorithms use a predictor pass to compute an estimate of the state at the $n+1/2$ frame rather than at the $n+1$ frame, as in conventional AM integration. The $n+1/2$ frame estimate is then used in the corrector pass to compute the $n+1$ state. The new RTAM (Real-time Adams-Moulton) algorithms require the external input U_n at the beginning of the first pass and $U_{n+1/2}$ at the beginning of the second pass, both of which can be made available as real-time inputs. Based on characteristic-root errors, sinusoidal transfer function gain and phase errors, and time-domain errors we have shown that the new RTAM-2, 3 and 4 integration algorithms are more accurate, respectively, than conventional AM-2, 3, and 4 algorithms. In addition they exhibit a somewhat larger region of stability in the zh plane. The paper also introduces two new three-pass real-time predictor-corrector algorithms, designated as P-2/PC-3/C-3 and P-3/PC-3/C-3, respectively. These new algorithms are shown to be considerably more accurate than RK-3 integration, based on characteristic root errors and time-domain response errors.

References

Benyon, P.K. 1968. "A Review of Numerical Methods for Digital Simulation," *SIMULATION*, 11(5):219-238.
 Gilbert, E.G. 1966. "Dynamic Error Analysis of Digital and Combined Digital Analog Systems," *SIMULATION*, 6(4):241-257.
 Gear, William G. 1971. *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
 Howe, R.M. 1985. "Transfer Function and Characteristic Root Errors for Fixed-Step Integration Algorithms," *Transactions of the Society for Computer Simulation*, Vol. 2, No. 4, December 1985, pp. 293-320.

ROBERT M. HOWE is a Professor of Aerospace Engineering at the University of Michigan in Ann Arbor, Michigan, where he has been a faculty member since receiving his PhD in Physics from M.I.T. in 1950. He teaches and conducts research in real-time simulation, flight dynamics, automatic control, and flight simulation. Dr. Howe served as first national chairman of Simulation Councils, Inc., the predecessor of SCS. In 1957 he was a "founder" of Applied Dynamics International, a simulation computer company based in Ann Arbor. In 1983 Dr. Howe received the AIAA deFlorez Training Award for Flight Simulation, and in 1988 The Society for Computer Simulation International Technical Award.

C++ Modeling & Simulation Class Library

Introduce Object Oriented Technology to your Engineering, Manufacturing, Finance or Forecasting Projects

Meijin++™

- ▶ Fundamental Structures
- ▶ Storable Data Collections
- ▶ Statistics & Math Classes
- ▶ Discrete Event Simulation
- ▶ F.F.T. & Differential Equations
- ▶ Exceptions & Debugging
- ▶ Time Series, Splines...

109 classes
 600 Page User's Manual
 C++ Documented Source
 No Royalties **\$279.00**

(714) 755-0995

Network Integrated Services, Inc.
 221 West Dyer Road
 Santa Ana CA 92707-3426
 Fax: (714) 433-2347