Apr 1st, 8:00 AM

Case Study in Digital Simulation

B. L. Capehart  
*Assistant Professor, ISE, University of Florida, GENESYS, Orlando, Florida*

R. M. Strong  
*Chief, Systems and Programming, Martin Marietta Corporation, Orlando, Florida*

Follow this and additional works at: [https://commons.erau.edu/space-congress-proceedings](https://commons.erau.edu/space-congress-proceedings)

Scholarly Commons Citation  
[https://commons.erau.edu/space-congress-proceedings/proceedings-1970-7th/session-12/3](https://commons.erau.edu/space-congress-proceedings/proceedings-1970-7th/session-12/3)
ABSTRACT

This paper presents a study of numerical methods for the simulation of continuous systems described by \( n \)-th order differential equations. The methods are applied to a wide spectrum of problems with the emphasis being on the practical rather than the theoretical results. The paper is directed toward the reader who is interested in comparative results of the application of various methods rather than derivations which can be found in a number of available texts.

The methods considered are Euler, modified Euler, classical Runge-Kutta, Milne's fourth-order predictor-corrector, Hamming's fourth-order predictor-corrector, a second-order predictor-corrector, Adams-Moulton, state variable, \( \Delta \) transform, and \( \Delta \) form. The methods are compared with respect to accuracy, computational efficiency, convenience of application and ease of programming.

The results of this case study should be helpful to the practicing engineer in selecting an appropriate digital simulation technique for his particular application.

INTRODUCTION

Digital simulation of continuous systems described by \( n \)-th order ordinary differential equations usually requires obtaining the solution of the differential equations by numerical approximation. Since any \( n \)-th order ordinary differential equation can be redefined as a system of first-order differential equations, we desire a particular numerical solution of the initial-value problem in which the differential equations are of the form

\[
\frac{dy}{dx} = f(x,y)
\]

and which passes through the given point \((x_0, y_0)\).

In this study we shall examine one-step methods, multistep methods, conversion to exact difference equation using \( \Delta \) transforms and conversion to an approximate difference equation using \( \Delta \) forms. All methods were evaluated on a CDC 6400 digital computer, using fixed step size.

The following notation will be used throughout the discussion of the methods and results:

- \( y_n \) = value of dependent variable after \( n \) steps
- \( f_n \) = value of derivative \( \frac{dy}{dx} \) at \( x_n \)
- \( h \) = increment in independent variable; i.e., \( x_{n+1} - x_n \)
- \( p_n \) = predicted value of dependent variable at step \( n \)
- \( p'_n \) = value of derivative of \( p_n \) evaluated at \( x_n \)
- \( c_n \) = corrected value of dependent variable at step \( n \)
- \( m_n \) = modified predicted value of dependent variable at step \( n \)
- \( m'_n \) = value of derivative of \( m_n \) evaluated at \( x_n \)
- \( y_{n,i} \) = value of dependent variable after \( i \)-th iteration at step \( n \)

One-step methods are procedures which depend only on the solution at \( x_n \) in order to produce the solution at \( x_{n+1} \) and are equivalent to initializing at each step in the process. Thus, simulating the system may be viewed as solving a sequence of initial-value problems, with the initial value for the current step being the solution of the previous process. Some methods (Runge-Kutta) do involve intermediate sub-steps within the current step. In a one-step method there is little difficulty varying the step size as the solution proceeds.

The one-step methods used in this study are:

- Euler's Method (EM), first-order
  \[
y_{n+1} = y_n + h f_n
\]
- Euler's Modified Method (EMM), second-order
  \[
p_{n+1} = y_n + h f_n \\
y_{n+1} = y_n + h f_n + \frac{h^2}{2} \left( f_n + f_{n+1} \right)
\]
- State Variable Method (SV)
  \[
y_{n+1} = e^{ah} y_n
\]
- Classical Runge-Kutta (RK), fourth-order
  \[
y_{n+1} = y_n + \frac{h}{6} \left[ k_1 + 2k_2 + 2k_3 + k_4 \right]
\]

\( k_1 = f(x_n, y_n) \\
k_2 = f \left( x_n + \frac{h}{2}, y_n + \frac{h}{2} k_1 \right) \\
k_3 = f \left( x_n + \frac{h}{2}, y_n + \frac{h}{2} k_2 \right) \\
k_4 = f \left( x_n + h, y_n + h k_3 \right)\]
\[ k_1 = h \cdot f(y_n, x_n) \]
\[ k_2 = h \cdot f(y_n + k_1/2, x_n + h/2) \]
\[ k_3 = h \cdot f(y_n + k_2/2, x_n + h/2) \]
\[ k_4 = h \cdot f(y_n + k_3, x_n + h) \]

Multistep methods utilize more of the information previously gained by some process in producing the solution at \( x_{n+1} \). Methods vary as to the number and use of the required solutions at \( x_{n+1}, x_n, x_{n-1}, x_{n-2}, \ldots \) A multistep procedure which requires only the solutions at \( x_n, x_{n-1}, x_{n-2}, \ldots \) in producing the solution at \( x_{n+1} \) is called a predictor method and is computed only once each step. A procedure which requires an estimate of the solution at \( x_{n+1} \) to produce the solution at \( x_{n+1} \) is called a corrector method. A predictor method is used to satisfy the corrector method's requisite for the solution at \( x_{n+1} \).

The combined use of a predictor method and a corrector method is referred to as a predictor-corrector system. The proper mating of predictor to corrector is important for efficiency. The predictor's sole function is to provide a good estimate of \( y_{n+1} \) and is computed only once. The corrector is normally iterated to meet some convergence criterion. Rapid convergence of the corrector depends on the initial estimate, hence a most desirable characteristic of the predictor is small truncation error. However, it is also advantageous to have the predictor and corrector possess similar order truncation error characteristics. The stability of the corrector is of prime importance and has received much attention. The correctors with smaller truncation error are probably more unstable, but this alone should not eliminate them from consideration for a particular problem. Some of the examples indicate the usefulness of such methods.

Multistep methods demand a one-step method to yield the required number of previous solution points. The entire set of previous solution points currently used are assumed to have been calculated with the current step size which is being used to obtain \( y_{n+1} \).

The multistep methods employed in the study are:

**Predictor-corrector using EMM, (PC), second-order**

\[ P_{n+1} = y_{n-1} + 3h \cdot f_n, \quad n \geq 0 \]
\[ y_{n+1} = y_n + \frac{h}{2} \cdot (f_n + p_{n+1}) \]

**Predictor-corrector using EMM and mop-up (PCM), second-order**

\[ P_{n+1} = y_{n-1} + 3h \cdot f_n, \quad n \geq 0 \]
\[ c_{n+1} = y_n + \frac{h}{2} \cdot (f_n + p_{n+1}) \]
\[ y_{n+1} = c_{n+1} + \frac{1}{3} (p_{n+1} - c_{n+1}) \]

**Milne predictor (MP), fourth-order**

\[ y_{n+1} = y_{n-3} + \frac{4h}{3} \cdot (2f_n - f_{n-1} + 2f_{n-2}) \]

**Milne predictor-corrector (M), fourth-order**

\[ P_{n+1} = y_{n-3} + \frac{4h}{3} \cdot (2f_n - f_{n-1} + 2f_{n-2}) \]
\[ y_{n+1} = y_n + \frac{h}{3} \cdot (p_{n+1} + 4f_n + f_{n-1}) \]

**Milne modified predictor-corrector (MM), fourth-order**

\[ P_{n+1} = y_{n-3} + \frac{4h}{3} \cdot (2f_n - f_{n-1} + 2f_{n-2}) \]
\[ m_{n+1} = p_{n+1} + \frac{28}{29} (y_n - y_{n+1}) \]
\[ y_{n+1} = m_{n+1} + \frac{3}{8} (p_{n+1} + 2f_n - f_{n-1}) \]

**Hamming predictor-corrector (H), fourth-order**

\[ P_{n+1} = y_{n-3} + \frac{4h}{3} \cdot (2f_n - f_{n-1} + 2f_{n-2}) \]
\[ y_{n+1} = \frac{1}{8} (9y_n - y_{n+1}) + \frac{3}{8} (p_{n+1} + 2f_n - f_{n-1}) \]

**Hamming modified predictor-corrector with mop-up (EMM), fourth-order**

\[ P_{n+1} = y_{n-3} + \frac{4h}{3} \cdot (2f_n - f_{n-1} + 2f_{n-2}) \]
\[ m_{n+1} = p_{n+1} + \frac{112}{121} (c_n - p_n) \]
\[ c_{n+1} = \frac{1}{8} (9y_n - y_{n+1}) + \frac{3}{8} (p_{n+1} + 2f_n - f_{n-1}) \]
\[ y_{n+1} = c_{n+1} + \frac{9}{121} (p_{n+1} - c_{n+1}) \]

**Adams-Moulton predictor-corrector (AM), fourth-order**

\[ P_{n+1} = y_n + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \]
\[ y_{n+1} = y_n + \frac{h}{24} \cdot (9p_n + 19f_{n+1} - 5f_{n-1} + f_{n-2}) \]

The above multistep methods used the classical fourth-order Runge-Kutta to produce the required starting points.

To examine the effects of iterating the corrector equation, the above methods and variations of those methods were computed both with and without convergence requirements on the corrector. When iteration of the corrector occurs the quantities \( p_{n+1} \) and \( m_{n+1} \) must be replaced by the derivative evaluated at the current result of the corrector equation. The predictor-corrector methods were also computed without evaluation of the final \( f_{n+1} \) after convergence. The convergence criterion applied in this study was

\[ \frac{y_{n+1,i+1} - y_{n+1,i}}{y_{n+1,i+1}} < \varepsilon \]

where \( \varepsilon \) is some small positive number the choice of which is influenced by the step size \( h \) and the estimated truncation error of the method employed.

Another pertinent but possibly misunderstood subject is the matter of convergence of the corrector. Iteration of the corrector under any convergence criterion is for the sole purpose of converging
the result of the corrector difference equation to the solution of that specific difference equation. It is not to be interpreted as convergence of the corrector difference equation to the true solution of the differential equation.

The $2$ transform method provides a procedure for converting a linear time-invariant differential equation into an exact difference equation. In the same fashion the $2$ form method can be used to produce an approximate difference equation by substituting for each $a^k$ a rational fraction in powers of $h^k$ into the place transform of the differential equation. These rational fractions can be found in tables for the $2$ forms. The difference equation for both the $2$ transform method and $2$ form method has the form

$$y_n = \sum_{j=0}^{M} a_{j+1} \delta_{nj} - \sum_{j=1}^{N} b_{j+1} y_{n-j}$$

where $Y(2) = \frac{a_1 + a_2 z^{-1} + \ldots + a_M z^{-M}}{1 + b_2 z^{-1} + \ldots + b_N z^{-N}}$ and $\delta_{nj}$ is the Kronecker delta.

**CASE STUDIES**

In the study six examples were run using all applicable methods. The complete computer results are not presented due to the volume of output. Only typical results at intermediate step sizes are given, the more accurate methods being accurate at smaller step sizes to about eleven significant digits in some of the problems.

The examples include linear time-invariant, linear time-variant, and non-linear differential equations. The linear time-invariant problems consist of both low-order and high-order differential equations with driving functions, and include a second-order "stiff" differential equation.

In all problems the step size remained fixed for all methods. The predictor-corrector methods were evaluated with many variations, using both a fixed number of iterations and a convergence criterion to terminate iteration of the corrector equation.

Example 1

$$y'' - y = \sin(3x)$$

The most accurate methods applied to this example were the state variable method and $2$ transform method. The former's accuracy is determined by the accuracy of $e^{Ah}$ while the $2$ transform results in an exact difference equation. The exact difference equation may be influenced by round-off as indicated by the results of this example. The difference equation is accurate to at least eleven significant digits from $h = .1$ to $h = .5$. With $h = .05$ the accuracy is comparable to the fourth-order methods. Decreasing the step size to $h = .001$, it is less accurate than the fourth-order methods are at the step size $h = .1$. The remaining remarks about this example exclude the SV and $2$ transform methods.

Although exhibiting considerable error, the $2$ form was the only stable method at $h = .5$. The $2$ form method was the most accurate method at the step sizes $h = .25$ and $h = .1$ and was still more accurate than DEW at $h = .01$. Nevertheless it had begun to display significant error at this step size.

With the step size $h = .25$, PC without mop-up was unstable. While PCM, with mop-up, did suffer from appreciable truncation error, it was stable. Milne's method was the most accurate, with HM being a close second. HM was at least $1\%$ more accurate than H and AM.

The results for $h = .1$ are shown in Figure 1. Of the predictor-correctors PC and PCM were effected the most by truncation errors, while again the mop-up in PCM yielded a significant improvement over PC in the results. Although HM was the most accurate method early in the process, it eventually suffers from more accumulated error. As the solution progressed both HM and AM were more accurate. The Hamming method H had more truncation error than any of the other fourth-order methods. Over the whole range of $x$ the ranking of the fourth-order methods relative to accuracy was HM, AM, MM, M, RK, and H, without iteration of the correctors.

At every step size, for all the fourth-order predictor-correctors, the results relative to the first corrector value degenerated due to any convergence requirement which caused iteration of the corrector equation. Figure 2 contains the results of iteration of the fourth-order predictor-corrector methods at $h = 0.1$. The information in this table demonstrates that, for this step size, the solution degenerates in this example due to iteration of the corrector even though convergence of the corrector is obtained. As the step size was decreased to $h = .05, .01$, and .001 similar consequences occurred although, as could be expected, the smaller step size diminished the effect of iteration. However, if allowed to iterate, M and MM converged to the same value, thus demonstrating that the predictor has no effect on the corrector when the corrector is iterated until convergence is attained.

Another interesting observation can be made after decreasing the step size to $h = .05$ and .01. For both step sizes HM became the most accurate method over the entire computed range of $x$, hence attesting to the usefulness of the mop-up calculation. The new ranking of fourth-order methods became HM, M, MM, RK, AM and H, where the corrector is not iterated. From this evidence the conclusion may be reached that, as the step size decreases, the truncation error estimation applied to the predicted value in the Milne's modified method is over- estimating the correction to be added to the predicted value. It has already been indicated that iteration of the corrector is not desirable for
this example when \( h = .1 \) and one expects the requirement for iteration to diminish as \( h \) decreases. Consequently it seems feasible that as the step size decreases, perhaps modification of the predicted value is not always beneficial if the step size is small enough that there is no need for iteration of the corrector.

At \( h = .001 \) modification of the predicted value essentially had no effect on the result. However, there is evidence at this step size that the modification of the corrector equation should be eliminated due to round-off contributions. At this step size there is a significant change in the ranking of the fourth-order methods. The new rank is \( M \geq MH, H, AM, RK, \) and \( HM \). \( HM \) was affected more quickly by round-off than the other fourth-order methods.

The Milne predictor was also computed at the same step sizes; its accuracy being better than PC and PCM but less than the other fourth-order methods.

**Example 2**

\[ x^2 y'' + xy' + x^2 y = 0 \]  \( \text{with } y(0) = 1 \text{ and } y'(0) = 0. \]

This is the Bessel equation of the first kind of order zero. Inasmuch as this is a time-variant differential equation, the \( t \) transform and \( t \) form methods were not applied. The state variable method was used on the problem to investigate the accuracy and efficiency of the method on this type problem. The transition matrix was evaluated at the mid-point of the computing interval. As anticipated, the results confirmed that the SV method should not be applied when the transition matrix must be evaluated each step. Truncating the exponential after a few terms or decreasing the step size in order to make the method competitive economically did not result in sufficient accuracy to warrant further consideration of the SV method on this type problem.

Unlike Example 1, in this example strict convergence requirements at larger step sizes improved the results of all the multistep methods. Iteration of the fourth-order correctors produced little effect at smaller step sizes of \( h \leq .05 \), again suggesting that any iteration requirement at smaller step sizes is inefficient.

This example does indicate the usefulness of iteration of the corrector on some problems in bringing stability to a multistep method at larger step sizes. Milne's methods were unstable at \( h = .5 \) without iteration of the corrector, but became stable with tighter convergence criterion. With \( h \leq .25 \), Milne's methods were stable without requiring iteration of the corrector.

In Figure 3 we see that truncation error still had significant influence on the results when the step size was reduced to \( h = .1 \). The convergence criterion resulted in an average of three iterations for the fourth-order predictor-correctors and four iterations for the second-order predictor-correctors. This step size resulted in \( HM \) becoming stable, even though there was considerable truncation error. PCM was again significantly more accurate than PC with convergence of the corrector improving both methods to a much greater degree than the other predictor-corrector methods. Without requiring convergence of the corrector, \( HM \) was more accurate than \( M \). Iteration of the corrector resulted in more improvement in \( M \) than in \( HM \), but not enough to overcome the advantage of the mop-up of the \( HM \) method. \( AM \) had greater truncation error than any other fourth-order method.

Smaller step sizes were run indicating, as in Example 1, that for the higher-order predictor-corrector methods both iteration of the corrector equation and the mop-up calculation are unnecessary and possibly undesirable at smaller step sizes. The mop-up computation produced negligible effect on the truncation error but did contribute to the round-off error of \( HM \).

**Example 3**

\[ y''' = y \text{ with } y(0) = y'(0) = 1. \]

The solution of this example is similar to the solution of Example 1 in that both are dominated by non-decreasing exponential terms. In this case, without iteration, the unmodified Milne method was more accurate for all step sizes than the method with the modification. Of course iteration resulted in convergence to the modified method's solution. Overall, the \( HM \) method was the best except at \( h = .001 \), where again it became the least accurate fourth-order method due to round-off.

**Example 4**

\[ y''' + 10y'' + 35y' + 50y = e^{-5x} \text{ with initial conditions } y(0) = y'(0) = y''(0) = 0. \]

In addition to simulating a higher-order differential equation, this example serves as a warning to be cautious in the selection of a numerical procedure on the basis of limited test ranges and step sizes. Milne's method gave excellent results until \( x = 2.5 \), when it suddenly became unstable for all step sizes.

Although Milne's method was unstable on this example, convergence of the corrector equation is possible. The example was computed using Milne's method with step size \( h = .1 \) and \( \epsilon = .001 \), requiring about six iterations for convergence to be obtained. This illustrates the point made earlier that convergence of the corrector does not validate the solution of the difference equation as representing the true solution.

The \( t \) transform solution was very accurate from \( h = .5 \) to \( h = .1 \), where round-off began to be noticeable.

This is the only example for which the \( RK \) was more accurate than all other approximation methods for each step size used and was the only stable method at \( h = .5 \). At \( h = .25 \) the methods PC, PCM, \( H \), and \( HM \) are unstable without iteration of the corrector.
Again when \( h \leq 0.05 \), the solutions degenerated with any iteration of the fourth-order corrector equations. In this example, the solution also degenerated with iteration of the PCM corrector.

**Example 5**

\[
y'' + 100.01 y' + 10y = 0 \quad \text{with} \quad y(0) = 0 \quad \text{and} \quad y'(0) = 1.
\]

This example shows that conventional numerical methods for solution of the initial-value problem with a large eigenvalue spread are not universally applicable. The 2 transform, being an exact difference equation, was the only accurate method and was able to maintain a high degree of accuracy over a large range of step sizes. The remaining methods were unstable for \( h > 0.001 \). However for step sizes \( h \leq 0.001 \), round-off error completely dominates the solution. The exception was the \( \Delta \) form which was not as subject to round-off, but the small step size required for stability prohibits efficient usage of the method on this type problem.

**Example 6**

\[
y'' + 20\sin(y) = 0 \quad \text{with} \quad y(0) = 0.5, \quad y'(0) = 0.
\]

The most important contribution of this example was that iteration of the corrector amplified any tendency of the solution of the corrector to oscillate at a particular solution point. This amplification was much more noticeable in those methods utilizing the mop-up calculation when compared to the corresponding methods without mop-up.

Again at larger step sizes iteration of the corrector brought stability to the Milne methods and to the PC and PCM methods. As the step size decreased the solution degenerated due to iteration of the corrector for all the predictor-correctors except PC.

Figure 4 shows the results for this example at \( h = 0.05 \) and with a convergence criterion that produced an average of two iterations per step for each fourth-order method, and three iterations for the second-order methods.

**EFFICIENCY OF METHODS**

Timing of computer runs on small problems such as those run in this study are not very dependable in comparative efficiency rating of the different methods. For example, the timing runs indicate that all the multistep methods required at least as much computation time as the Runge-Kutta. A knowledgeable analyst knows this should not be the case. Small problems and programming inefficiencies accentuate the overhead cost relative to the evaluation of the derivative, whereas in large simulations the cost of calculating the derivatives overshadows the overhead computation involved in employing the numerical method used to solve the system of differential equations.

On the problems used in this study, Figure 5 indicates the ratio of the computer time to that of EM for \( h = 0.05 \) for Example 2. The predictor-corrector methods were run without iteration. This table is not relative to equivalent accuracy, and is presented only to indicate relative overhead computation for each method.

The multistep methods are potentially capable of more efficiency relative to equivalent accuracy because they utilize more of the available information about the solution. The fourth-order predictor-corrector methods are too close to being equivalent in accuracy and computation time to compare relative computational efficiencies under the constraints used in this study.

Looking at Figure 6 we see a comparison of the number of derivative evaluations for comparable accuracy for several methods on Example 1. Here the potential efficiency of the high-order multistep methods such as the Adams-Moulton method and Hanning’s methods can be seen.

It should be noted also that computations were made for the multistep methods to show that as the step size decreased the evaluation of the derivative after the final application of the corrector is not necessary unless the corrector value is modified as in PCM and EM. Even then the derivative need not be evaluated at \( y_{n+1} \) if \( y_{n+1} \) is close to \( y_{n+1} \).

**OBSERVATIONS ABOUT FIXED STEP SIZE PROCEDURES**

After examining a large amount of computational data the conclusion was reached that if a fixed step procedure is being used, the maximum number of iterations should be no greater than three. For higher-order predictor-corrector methods, perhaps the limit should be two. It has been demonstrated in this paper that iteration of the corrector to convergence is not assurance of more accurate results for every problem; therefore stricter convergence requirements may not be the desirable approach to a more accurate solution.

A more significant reduction in truncation error and increased stability is made by a reduction in step size \( h \). In the examples presented, computed results after any number of iterations were significantly less accurate than halving the step size with no iteration. Except for computational overhead of the method, no iteration at half the step size is at worst as costly as any iteration at \( h \), with the potential of significant improvement in efficiency. This is particularly true when the evaluation of the derivative dominates the time required for an iteration, thus making overhead insignificant.

In Example 2 and Example 6 some of the predictor-corrector methods were unstable at larger step sizes without iteration. In this situation, although the method was stable with more strict convergence criterion, truncation error was prominent. In some of the test problems iteration improved the results relative to initial corrector evaluation.
at one point but caused the solution to degenerate at others. In general one cannot forecast the quality of the effect of iteration of the corrector.

When the mop-up equation was used on predictor-corrector PCM in Example 4, iteration of the corrector caused degeneration of the solution at any step size. The evidence indicates that on some problems the corrector should not be iterated when mop-up is used on the corrector.

Caution must be used in the selection of both the step size and the convergence factor $\xi$ which assumes a priori estimation of the truncation error per step. Improper choice of the step size can result in instability, round-off and truncation error, as well as causing the corrector to iterate unnecessarily. The convergence factor $\xi$ can result in instability if too large and can cause the maximum number of iterations to result if too small.

**PROGRAMMING CONSIDERATIONS**

The relative difficulty of programming the methods seems in general to be directly proportional to the order of the method. The procedures required to program any of the methods is straightforward for most methods, with the multistep methods having the added nuisances of demanding a one-step method as a starter and requiring updating of more previous solutions. Assuming the existence of a one-step method, the multistep methods, regardless of the order, require an equivalent amount of effort. The multistep methods require less effort than programming the Runge-Kutta which required the most effort of any of the methods employed. In fact, with little additional labor the routine can have several of the multistep methods as options to offer selection of the best method for the individual problem.

The easiest methods to program are the Euler methods. The state variable method is comparable to the Euler methods in programming effort if a routine to evaluate the exponential $e^{AH}$ and a matrix product routine already exist. The state variable method requires comparable effort to the multistep methods if the routine to evaluate $e^{AH}$ must be generated.

The effort to program the $Z$ transform and the $Z$ form methods is dependent upon the external calculations required. If only the solution of the difference equation is programmed with the coefficients of the difference equation and starting values being input, the $Z$ transform and $Z$ form methods are slightly more difficult than the Euler methods. However, if the calculation of the coefficients and starting values are performed in the computer program, the $Z$ transform and $Z$ form methods are likewise comparable to the multistep methods in programming effort.

**COMPARATIVE RESULTS**

The results of the example system simulations provide the basis for the following recommendations of numerical technique selection. Since only a limited number of numerical methods on a small sample of problems were investigated, these recommendations are based on what appear to be the most consistent methods of those examined. These recommendations are influenced to some extent by the nature of the systems being simulated, since several of the methods are not suitable for simulating general systems.

For linear time-invariant systems the $Z$ transform was the most efficient method for comparable accuracy. The method can be affected significantly by round-off at very small step sizes. The $Z$ form method is not as efficient for comparable accuracy as the $Z$ transform and state variable methods. At certain step sizes the $Z$ form method was more efficient than any of the other methods on several of the examples. The generation of the difference equation using the $Z$ transform or $Z$ form can be quite an exercise in algebra leading to many opportunities for error, so the methods are not recommended for systems of higher than third order.

The state variable method is the most efficient for high-order linear time-invariant systems when the $Z$ transform is not easily attainable. For such systems the exponential $e^{AH}$ is evaluated only once.

The Hamming method with the final correction for the corrector was the most consistent performer for the simulation of general systems. Although Milne's method has less truncation error on some problems at certain step sizes and has better round-off properties, Hamming's methods are a better compromise between stability and truncation error. The mop-up computation was a significant improvement in Hamming's method except at very small step sizes when round-off became significant. In such instances the mop-up computation should not be used.

The Hamming method without the mop-up computation and Adams-Moulton were comparable with respect to accuracy and computational efficiency. The modification of the predictor equation used in HM may be used effectively in the Hamming's method $H$.

Milne's method was the most accurate method on some of the problems. The method was unstable on others. Although the method is unstable on some problems, this alone should not eliminate it from consideration for a particular problem. Milne's predictor exhibited more truncation error than any of the other fourth-order methods but was more accurate and more stable than the second-order predictor-corrector PC and PCM.

The second-order predictor-correctors were not as efficient as any of the fourth-order methods. They can be quite useful in obtaining reasonable results during the checkout stage of a simulation. Of the multistep methods, they present the least difficulty in varying step sizes. The mop-up computation in PCM produced a significant improvement in the results over that produced by PC.
The Runge-Kutta was the only one-step method suitable for general simulations. The RK method was not as efficient as the other fourth-order methods examined in this study. The method is advantageous to have available as a starter to generate "initial" values for multistep methods. In fact, the method should normally be used only as a starter method for multistep methods.

The remaining one-step methods EM and EMM are too inefficient for comparable accuracy to be considered for anything other than rough approximations.

The reader is reminded of the constraints under which this study was conducted; constraints which might prejudice some of the conclusions reached. Using a fixed step size for the methods is not the optimum procedure for comparison of the methods, since variable step procedures should be considered. Also, forcing a prescribed average number of iterations of the corrector equation, as done for Figure 2, is not recommended as the stability of the corrector may be adversely affected. Convergence criterion was used for all the predictor-corrector methods, but proper choice of \( \xi \) can effect a prescribed number of iterations. This procedure was used to generate Figure 2.

**CONCLUSION**

The primary emphasis of this study has been to investigate the application of a number of numerical techniques to the simulation of continuous systems described by n-th order differential equations. The results of the study are influenced by the particular set of example systems chosen, and the recommendations made regarding specific methods are based on these results. However, even with this limitation, the results of this case study should be helpful to the practicing engineer in selecting an appropriate digital simulation technique for his particular application.

**REFERENCE**


**BIBLIOGRAPHY**


<table>
<thead>
<tr>
<th>X</th>
<th>1.0</th>
<th>3.0</th>
<th>6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>.407628</td>
<td>23.2428</td>
<td>9389.75</td>
</tr>
<tr>
<td>PC</td>
<td>.408672</td>
<td>23.6044</td>
<td>9728.99</td>
</tr>
<tr>
<td>PCM</td>
<td>.407690</td>
<td>23.2417</td>
<td>9389.07</td>
</tr>
<tr>
<td>MP</td>
<td>.407629</td>
<td>23.2437</td>
<td>9389.40</td>
</tr>
<tr>
<td>M</td>
<td>.407602</td>
<td>23.2437</td>
<td>9388.98</td>
</tr>
<tr>
<td>MM</td>
<td>.407624</td>
<td>23.2435</td>
<td>9390.55</td>
</tr>
<tr>
<td>H</td>
<td>.407612</td>
<td>23.2409</td>
<td>9388.33</td>
</tr>
<tr>
<td>HM</td>
<td>.407628</td>
<td>23.2428</td>
<td>9389.75</td>
</tr>
<tr>
<td>AM</td>
<td>.407628</td>
<td>23.2428</td>
<td>9389.75</td>
</tr>
<tr>
<td>RK</td>
<td>.407628</td>
<td>23.2428</td>
<td>9389.75</td>
</tr>
<tr>
<td># form</td>
<td>.407628</td>
<td>23.2428</td>
<td>9389.75</td>
</tr>
</tbody>
</table>

Figure 1. Example 1 with step size h = .1.

<table>
<thead>
<tr>
<th>X</th>
<th>2.0</th>
<th>5.0</th>
<th>9.0</th>
<th>14.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>.2238908</td>
<td>-.1775968</td>
<td>-.090336</td>
<td>.1710735</td>
</tr>
<tr>
<td>EM</td>
<td>.2271576</td>
<td>-.1942548</td>
<td>-.1577351</td>
<td>.3053454</td>
</tr>
<tr>
<td>EMM</td>
<td>.2326925</td>
<td>-.1775932</td>
<td>-.0928647</td>
<td>.1676548</td>
</tr>
<tr>
<td>PC</td>
<td>.2244250</td>
<td>-.1766448</td>
<td>-.0888995</td>
<td>.1725806</td>
</tr>
<tr>
<td>PCM</td>
<td>.2239158</td>
<td>-.1776084</td>
<td>-.0902920</td>
<td>.1710484</td>
</tr>
<tr>
<td>M</td>
<td>.2238906</td>
<td>-.1775963</td>
<td>-.0903344</td>
<td>.1710724</td>
</tr>
<tr>
<td>MM</td>
<td>.2238906</td>
<td>-.1775962</td>
<td>-.0903343</td>
<td>.1710721</td>
</tr>
<tr>
<td>H</td>
<td>.2238908</td>
<td>-.1775967</td>
<td>-.0903336</td>
<td>.1710731</td>
</tr>
<tr>
<td>HM</td>
<td>.2238908</td>
<td>-.1775967</td>
<td>-.0903336</td>
<td>.1710731</td>
</tr>
<tr>
<td>AM</td>
<td>.2238908</td>
<td>-.1775967</td>
<td>-.0903336</td>
<td>.1710731</td>
</tr>
<tr>
<td>RK</td>
<td>.2238910</td>
<td>-.1775978</td>
<td>-.0903327</td>
<td>.1710754</td>
</tr>
</tbody>
</table>

Figure 3. Example 2 with step size h = .1.

<table>
<thead>
<tr>
<th>X</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
<th>5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact</td>
<td>-.153282</td>
<td>-.407639</td>
<td>.401602</td>
<td>.163089</td>
<td>-.499895</td>
</tr>
<tr>
<td>EM</td>
<td>-.330662</td>
<td>-.697599</td>
<td>1.80882</td>
<td>-.254016</td>
<td>-.5.97884</td>
</tr>
<tr>
<td>EMM</td>
<td>-.333133</td>
<td>-.430708</td>
<td>.377961</td>
<td>.227002</td>
<td>-.507171</td>
</tr>
<tr>
<td>MP</td>
<td>-.153636</td>
<td>-.407141</td>
<td>.402290</td>
<td>.161508</td>
<td>-.499923</td>
</tr>
<tr>
<td>PO</td>
<td>-.161292</td>
<td>-.397455</td>
<td>.416711</td>
<td>.129496</td>
<td>-.499217</td>
</tr>
<tr>
<td>PCM</td>
<td>-.153826</td>
<td>-.409607</td>
<td>.405417</td>
<td>.163321</td>
<td>-.506798</td>
</tr>
<tr>
<td>M</td>
<td>-.153261</td>
<td>-.407668</td>
<td>.401553</td>
<td>.165193</td>
<td>-.498899</td>
</tr>
<tr>
<td>MM</td>
<td>-.153261</td>
<td>-.407669</td>
<td>.401554</td>
<td>.165193</td>
<td>-.499899</td>
</tr>
<tr>
<td>H</td>
<td>-.153174</td>
<td>-.407885</td>
<td>.401446</td>
<td>.165698</td>
<td>-.500095</td>
</tr>
<tr>
<td>HM</td>
<td>-.153268</td>
<td>-.407601</td>
<td>.401512</td>
<td>.165110</td>
<td>-.499742</td>
</tr>
<tr>
<td>AM</td>
<td>-.153187</td>
<td>-.407826</td>
<td>.401438</td>
<td>.165993</td>
<td>-.499995</td>
</tr>
<tr>
<td>RK</td>
<td>-.153318</td>
<td>-.407980</td>
<td>.401652</td>
<td>.162932</td>
<td>-.499859</td>
</tr>
</tbody>
</table>

Figure 4. Example 6 with step size h = .05.
<table>
<thead>
<tr>
<th>Method</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM</td>
<td>1.00</td>
</tr>
<tr>
<td>EMM</td>
<td>1.41</td>
</tr>
<tr>
<td>PC</td>
<td>1.50</td>
</tr>
<tr>
<td>PCM</td>
<td>1.97</td>
</tr>
<tr>
<td>M</td>
<td>2.14</td>
</tr>
<tr>
<td>RK</td>
<td>2.17</td>
</tr>
<tr>
<td>H</td>
<td>2.24</td>
</tr>
<tr>
<td>MM</td>
<td>2.42</td>
</tr>
<tr>
<td>AM</td>
<td>2.66</td>
</tr>
<tr>
<td>HM</td>
<td>2.72</td>
</tr>
</tbody>
</table>

Figure 5. Ratio of computer time for Example 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>RK</th>
<th>PC</th>
<th>PCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>step size</td>
<td>.1</td>
<td>.01</td>
<td>.025</td>
</tr>
<tr>
<td>number of iterations</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>number of derivative evaluations</td>
<td>4</td>
<td>20</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>HM</th>
<th>PC</th>
<th>PCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>step size</td>
<td>.1</td>
<td>.002</td>
<td>.02</td>
</tr>
<tr>
<td>number of iterations</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>number of derivative evaluations</td>
<td>2</td>
<td>100</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 6. Derivative evaluations per step for comparable accuracy for Example 1.