APPENDIX I - NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS

"Computers are influencing network theory by demanding methods of analysis adapted to the solution of computer-sized problems," as stated by F.H. Branin [2], but very little of this influence has shown up yet in textbooks on electric circuits and networks, not even in most of the recently published books. In this appendix, an attempt is made to summarize some of the numerical solution techniques for solving ordinary differential equations, which one might consider in developing a general-purpose program, such as the EMTP. Since power system networks are mostly linear, techniques for linear ordinary differential equations are given special emphasis.

I.1 Closed Form Solution

Let us assume that the differential equations are written in "state-variable form," and that the equations are linear,

$$\left[\frac{dx}{dt}\right]=[A][x]+[g(t)], \tag{I.1}$$

with a constant square matrix [A], and a vector of known forcing functions [g(t)]. There is no unique way of writing equations in state variable form, but it is common practice to choose currents in inductances and voltages across capacitances as state variables. For example, Eq. (I.1) could have the following form for the network of Fig. I.1:

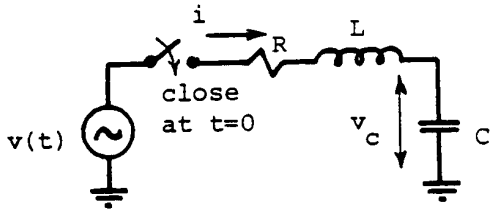

Fig. I.1 - Energization of an R-L-C network

$$\begin{bmatrix}\frac{di}{dt}\\ \frac{dv_c}{dt}\end{bmatrix}=\begin{bmatrix}-\frac{R}{L} & -\frac{1}{L}\\ \frac{1}{C} & 0\end{bmatrix}\begin{bmatrix}i\\ v_c\end{bmatrix}+\begin{bmatrix}\frac{1}{L}v(t)\\ 0\end{bmatrix} \tag{I.2}$$

With Laplace transform methods, especially when one output is expressed as a function of one input, the system is often described as one n^{th}-order differential equation, e.g., for the example of Fig. I.1 in the form

Such an n^{th}-order differential equation can of course always be rewritten as a system of n first-order differential equations, by introducing extra variables $x_2 = dx_1/dt$, $x_3 = dx_2/dt$, to $x_n = dx_{n-1}/dt$, for the higher-order derivatives,

with $x_1 = x$. In the example, with $x_1 = i$ and $x_2 = di/dt$,

$$\begin{bmatrix} R & L \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{C} & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{dv(t)}{dt} \\ 0 \end{bmatrix}$$

which, after pre-multiplication with

$$\begin{bmatrix} R & L \\ 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 \\ \frac{1}{L} & -\frac{R}{L} \end{bmatrix}$$

produces another state-variable formulation for this example. While [A] of this formulation differs from that in Eq. (I.2), its eigenvalues are the same.

The closed-form solution of Eq. (I.1), which carries us from the state of the system at t - Δt to that at t, is

$$[x(t)] = e^{[A]\Delta t} \cdot [x(t-\Delta t)] + \int_{t-\Delta t}^{t} e^{[A](t-u)} [g(u)] du, \tag{I.3}$$

where the matrix $e^{[A]\Delta t}$ is called the "transition matrix." Eq. (I.3) contains the case where [x(t)] is simply desired as a function of t by setting Δt = t. The computational task lies in finding this transition matrix. Since there is no closed-form solution for the matrix exponential $e^{[A]\Delta t}$, the way out is to transform this matrix to a diagonal matrix, whose elements can easily be evaluated by using the eigenvalues λ_i of [A] and the matrix of eigenvectors (modal matrix) [M] of [A]. and then to transform back again. An efficient method for finding eigenvalues appears to be the "QR transformation" due to J.G.F. Francis [3], and for finding eigenvectors the "inverse iteration scheme" due to J.H. Wilkinson [4], which has also been described in modified form by J.E. Van Ness [5]. With [Λ] and [M] known, where [Λ] is the diagonal matrix of eigenvalues λ_i, $e^{[A]\Delta t}$ is diagonalized[1],

$$[M]^{-1} e^{[A]\Delta t} [M] = [e^{\Lambda\Delta t}]$$

Once the diagonal elements $e^{\lambda_i \Delta t}$ have been found, this can be converted back to give

$$e^{[A]\Delta t} = [M]\ [e^{\Lambda\Delta t}]\ [M^{-1}] \tag{I.4}$$

where

$[e^{\Lambda\Delta t}]$ = diagonal matrix with elements $e^{\lambda_i \Delta t}$,

[M] = eigenvector (modal) matrix of [A], and

---

[1]If [M] diagonalizes [A], it will also diagonalize $e^{[A]\Delta t}$. The matrix exponential is defined as the series of Eq. (I.13), and then one simply has to show that [M] not only diagonalizes [A], but all positive powers $[A]^n$ as well. Since $[A] = [M][\Lambda][M]^{-1}$ it follows that $[A]^n = ([M][\Lambda][M]^{-1})([M][\Lambda][M]^{-1})\ldots([M][\Lambda][M]^{-1})$ of $[A]^n = [M][\Lambda^n][M]^{-1}$. Therefore, $[M]^{-1}[A]^n[M] = [\Lambda^n]$ is again diagonal.

λ_i = eigenvalues of [A].

With Eq. (I.4), Eq. (I.3) becomes

$$[x(t)] = [M]\ [e^{\Lambda\Delta t}]\ [M]^{-1}[x(t-\Delta t)] + \int_{t-\Delta t}^{t} [M]\ [e^{\Lambda(t-u)}]\ [M]^{-1}\ [g(u)]\ du \qquad \text{(I.5)}$$

The "convolution integral" in Eq. (I.5) can be evaluated in closed form for many types of functions [g(t)].

For the network of Fig. I.1, the eigenvalues can be obtained by setting the determinant of [A]-λ[U] to zero ([U] = identity matrix),

$$\begin{vmatrix} -\frac{R}{L} - \lambda & -\frac{1}{L} \\ \frac{1}{C} & -\lambda \end{vmatrix} = 0$$

or

$$\lambda_{1,2} = -\frac{R}{2L} \pm \sqrt{(\frac{R}{2L})^2 - \frac{1}{LC}} \qquad \text{(I.6)}$$

If $R < 2\sqrt{(L/C)}$, then the system is underdamped[2], and the argument under the square root will be negative, giving a pair of complex eigenvalues

$$\lambda_{1,2} = \alpha \pm j\beta; \quad with \quad \alpha = -\frac{R}{2L}, \quad \beta = \sqrt{\frac{1}{LC} - (\frac{R}{2L})^2} \qquad \text{(I.7)}$$

For a specific case, let us assume that R = 1Ω, L = 1H, C = 1F. then

$$\lambda_{1,2} = -\frac{1}{2} \pm j\frac{\sqrt{3}}{2} = e^{\pm j120^\circ}$$

and

$$[M] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ e^{-j120^\circ} & e^{j120^\circ} \end{bmatrix}, \qquad [M]^{-1} = \frac{\sqrt{2}}{j\sqrt{3}} \begin{bmatrix} e^{j120^\circ} & -1 \\ -e^{-j120^\circ} & 1 \end{bmatrix}$$

If we set Δt = t to obtain the state variables simply as a function of time and of initial conditions, then Eq. (I.5) becomes

---

[2]If R > 2*sqrt(L/C), then the system is overdamped, giving two real eigenvalues. The critically damped case of R = 2 * sqrt(L/C) seldom occurs in practice; it leads to two identical eigenvalues. This latter case of "multiple eigenvalues" may require special treatment, which is not discussed here.

$$\begin{bmatrix} i(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} e^{\alpha t}\left(\cos\beta t - \frac{1}{\sqrt{3}}\sin\beta t\right) & -\frac{2}{\sqrt{3}}e^{\alpha t}\sin\beta t \\ \frac{2}{\sqrt{3}}e^{\alpha t}\sin\alpha t & e^{\alpha t}\left(\cos\beta t + \frac{1}{\sqrt{3}}\sin\beta t\right) \end{bmatrix} \cdot \begin{bmatrix} i(0) \\ v_c(0) \end{bmatrix}$$
$$+ \frac{2}{\sqrt{3}} \int_0^t \begin{bmatrix} e^{\alpha(t-u)}\left[\cos(\beta(t-u)) - \frac{1}{\sqrt{3}}\sin(\beta(t-u))\right]v(u) \\ e^{\alpha(t-u)}\sin(\beta(t-u))v(u) \end{bmatrix} du \qquad \text{(I.8)}$$

with α and β as defined in Eq. (I.7). If we were to assume that the voltage source is zero and that $v_c(0) = 1.0$ p.u., then we would have the case of discharging the capacitor through R-L, and from Eq. (I.8) we would immediately get (realizing that i(0) = 0),

$$i(t) = -\frac{2}{\sqrt{3}}\, e^{\alpha t}\, \sin\beta t$$

$$v_c(t) = e^{\alpha t}\left(\cos\beta t + \frac{1}{\sqrt{3}}\sin\beta t\right)$$

Could such a closed-form solution be used in an EMTP? For networks of moderate size, it probably could. J.E. Van Ness had no difficulties finding eigenvalues and eigenvectors in systems of up to 120 state variables [5]. If the network contains switches which frequently change their position, then its implementation would probably become very tricky. Combining it with Bergeron's method for distributed-parameter lines, or with more sophisticated convolution methods for lines with frequency-dependent parameters, should in principle be possible. Where the method becomes almost unmanageable, or useless, is in networks with nonlinear elements. Another difficulty would arise with the state-variable formulation, because Eq. (I.1) cannot as easily be assembled by a computer as the node equations used in the EMTP. This difficulty could be overcome, however, since there are ways of using node equations even for state-variable formulations, by distinguishing node types according to the types of branches (R, L, or C) connected to them.

Where do Laplace transform methods fit into this discussion since they provide closed-form solutions as well? To quote F.H. Branin [2], "…traditional methods for hand solution of networks are not necessarily best for use on a computer with networks of much greater size. the Laplace transform techniques fit this category and should at least be supplemented, if not supplanted, by numerical methods better adapted to the computer>" He then goes on to show that essentially all of the information obtainable by Laplace transforms is already contained in the eigenvalues and eigenvectors of [A]. It is surprising that very few, if any, textbooks show this relationship. The Laplace transform of Eq. (I.1) is

$$s[X(s)] - [x(0)] = [A]\cdot[X(s)] + [G(s)] \qquad \text{(I.9a)}$$

or rewritten

$$(s\,[U] - [A]) \cdot [X(s)] = [x(0) + [G(s)] \tag{I.9b}$$

From which the formal solution in the s-domain is obtained as

$$[X(s)] = (s[U] - [A])^{-1} \cdot ([x(0)] + [G(s)]) \tag{I.10}$$

The computational task in Eq. (I.10) is the determination of the inverse of (s[U]-[A]). The key to doing this efficiently is again through the eigenvalues and eigenvectors of [A]. With that information, the matrix (s[U]-[A]) is diagonalized,

$$[M]^{-1} \cdot (s\,[U] - [A]) \cdot [M] = s\,[U] - [\Lambda] \tag{I.11}$$

and then the inverse becomes

$$(s\,[U] - [A])^{-1} = [M] \cdot (s\,[U] - \Lambda)^{-1} \cdot [M]^{-1} \tag{I.12}$$

in which the inverse on the right-hand side i now trivial to calculate since (s[U]-[Λ]) is a diagonal matrix (that is, one simply takes the reciprocals of the diagonal elements). To quote again from F.H. Branin [2], "…one of the more interesting features of this method is the fact that it is far better suited for computer-sized problems than the traditional Laplace transform techniques involving ratio of polynomials and the poles and zeros thereof. In particular, the task of computing the coefficients of the polynomials in a network function P(s)/Q(s) is not only time-consuming but also prone to serious numerical inaccuracies, especially when the polynomials are of a high degree. The so-called "topological" formula approach [25] to computing these network functions involves finding all the trees of a network and then computing the sum of the corresponding tree-admittance products. But the number of trees may run into millions for a network with only 20 nodes and 40 branches. And even if this were not enough of an impediment, the computation of the roots of the polynomials P(s) and Q(s) is hazardous because these roots may be extremely sensitive to errors in the coefficients. In the writer's judgment, therefore, the polynomial approach is just not matched to the network analysis tasks which the computer is called upon to handle. The eigenvalue approach is much better suited and gives all of the theoretical information that the Laplace transform methods are designated to provide. For example, the eigenvalues are identical with the poles of the network functions. Moreover, any network function desired may be computed straightforwardly and its sensitivity obtained, either with respect to frequency or with respect to any network parameter. Finally, even the pole sensitivities can be calculated…"

I.2 Taylor Series Approximation of Transition Matrix

The matrix exponential $e^{[A]\Delta t}$ can be approximated by a power series, derived from a Taylor series expansion,

$$e^{[A]\Delta t} = [U] + \Delta t\cdot[A] + \frac{\Delta t^2}{2!}[A]^2 + \frac{\Delta t^3}{3!}[A]^3 + \frac{\Delta t^4}{4!}[A]^4 + \ldots \qquad \text{(I.13)}$$

This series is, in effect, the definition of the matrix exponential.

Using Eq. (I.13), necessarily with a finite number of terms, appears to offer a way around the computation of eigenvalues. However, "the method runs headlong into another kind of eigenvalue problem which limits its usefulness: namely, that when the matrix [A] has a large eigenvalue (which means a small time constant), the integration step Δt must be kept small in order to permit rapid convergence of Eq. (I.13)" [2]. This refers to the problem encountered in "stiff systems", where there are large differences between the magnitudes of eigenvalues, and where the largest eigenvalues produce "ripples" of little interest to the engineer, who is more interested in the slower changes dictated by the smaller eigenvalues, as indicated in Fig. I.2. The method of using Eq. (I.13) becomes numerically unstable, for a given finite

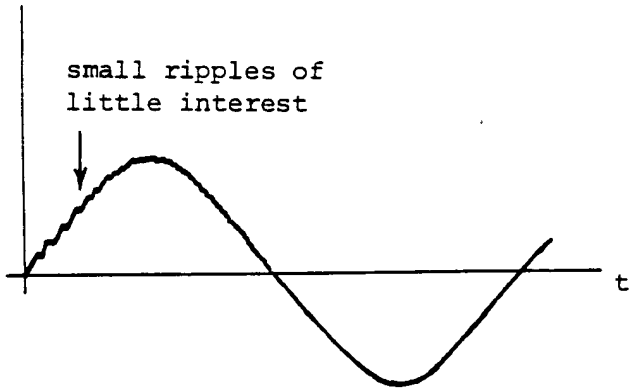

Fig. I.2 - Response of a "stiff system"

number of terms if Δt is not sufficiently small to trace the small, uninteresting ripples. It is, therefore, not a practical method for an EMTP. It exhibits the same proneness to numerical instability as the Runge-Kutta method discussed in Section I.5, which is not too surprising, since this method becomes identical with the fourth-order Runge-Kutta method if 5th and higher-order terms are neglected in Eq. (I.13), at least if the forcing function [g(t)] in Eq. (I.1) is zero ("autonomous system"), as further explained in Section I.5. Since this method is not practical, more details such as the handling of the convolution integral in Eq. (I.3) are not discussed.

I.3 Rational Approximation of Transition Matrix

A rational approximation for the matrix exponential, which is numerically stable and therefore much better than Eq. (I.13), is due to E.J. Davison[3] [6],

[3]This was pointed out to the writer by K.N. Stanton when he was at Purdue University (now President of ESCA Corp. in Seattle)

$$e^{[A]\Delta t} \approx \left([U] - \frac{\Delta t}{2}[A] + \frac{\Delta t^2}{4}[A]^2 - \frac{\Delta t^3}{12}[A]^3 \right)^{-1} \cdot \left([U] + \frac{\Delta t}{2}[A] + \frac{\Delta t^2}{4}[A]^2 + \frac{\Delta t^3}{12}[A]^3 \right) \tag{I.14}$$

A lower-order rational approximation, which is also numerically stable for all Δt, neglects the second and high-order terms in Eq. (I.14).

$$e^{[A]\Delta t} \approx \left([U] - \frac{\Delta t}{2}[A] \right)^{-1} \cdot \left([U] + \frac{\Delta t}{2}[A] \right) \tag{I.15}$$

This is identical with the trapezoidal rule of integration discussed in the following section.

Would it be worthwhile to improve the accuracy of the EMTP, which now uses the trapezoidal rule, with the higher-order rational approximation of Eq. (I.14)? This is a difficult question to answer. First of all, the EMTP is not based on state-variable formulations, and it is doubtful whether this method could be applied to individual branch equations as easily as the trapezoidal rule (see Section 1). Furthermore, if sparsity is to be exploited, much of the sparsity in [A] could be destroyed when the higher-order terms are added in Eq. (I.14). By and large, however, the writer would look favorably at this method if the objective is to improve the accuracy of EMTP results, even though it is somewhat unclear how to handle the convolution integral in Eq. (I.3).

I.4 Trapezoidal Rule of Integration

Since this is the method used in the EMTP, the handling of the forcing function [g(t)] in Eq. (I.1), or analogously the handling of the convolution integral in Eq. (I.3), shall be discussed here. Let Eq. (I.1) be rewritten as an integral equation,

$$[x(t)] = [x(t-\Delta t)] + \int_{t-\Delta t}^{t} \left([A]\,[x(u) + [g(u)] \right) du \tag{I.16}$$

which is still exact. By using linear interpolation on [x] and [g] between t-Δt and t, assuming for the time being that [x] were known at t (which, in reality, is not true, thereby making the method "implicit"), we get

$$[x(t)] = [x(t-\Delta t)] + \frac{\Delta t}{2}[A] \cdot \left([x(t-\Delta t)] + [x(t)] \right) + \frac{\Delta t}{2} \cdot \left([g(t-\Delta t)] + [g(t)] \right) \tag{I.17}$$

Linear interpolation implies that the areas under the integral of Eq. (I.16) are approximated by trapezoidals (Fig. I.3); therefore the name "trapezoidal rule of integration". The method is identical with using "central difference quotients" in Eq. (I.1),

$$\frac{[x(t)] - [x(t-\Delta t)]}{\Delta t} = [A] \frac{[x(t-\Delta t)] + [x(t)]}{2} + \frac{[g(t-\Delta t)] + [g(t)}{2} \qquad \text{(I.18)}$$

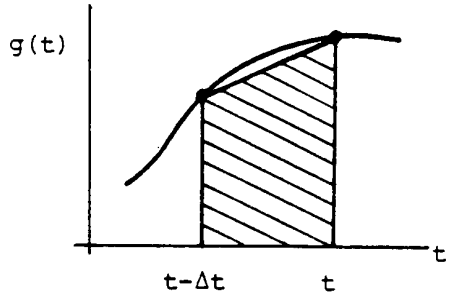

Fig. I.3 - Trapezoidal rule of integration

and could just as well be called the "method of central difference quotients". Eq. (I.17) and (I.18) can be rewritten as

$$\left([U] - \frac{\Delta t}{2}[A] \right) \cdot [x(t)] = \left([U] + \frac{\Delta t}{2}[A] \right) \cdot [x(t-\Delta t)] + \frac{\Delta t}{2} \left(g(t-\Delta t)] + [g(t)] \right) \qquad \text{(I.19)}$$

which, after premultiplication with $([U]-\Delta t \bullet [A]/2)^{-1}$, shows that we do indeed get the approximate transition matrix of Eq. (I.15).

Working with the trapezoidal rule of integration requires the solution of a system of linear, algebraic equations in each time step. If Δt is not changed, and as long as no network modifications occur because of switching or nonlinear effects, the matrix $([U]-\Delta t \bullet [A]/2)$ for this system of equations remains constant. It is therefore best and most efficient to triangularize this matrix once at the beginning, and again whenever network changes occur, and to perform the downward operations and backsubstitutions only for the right-hand side inside the time step loop, using the information contained in the triangularized matrix. The solution process is broken up into two parts in this scheme, one being the triangularization of the constant matrix, the other one being the "repeat solution process" for right-hand sides (which is done repeatedly inside the time step loop). this concept of splitting the solution process into one part for the matrix and a second part for the right-hand side is seldom mentioned in textbooks, but it is very useful in many power system analysis problems, not only here, but also in power flow iterations using a triangularized [Y]-matrix, as well as in short-circuit calculations for generating columns of the inverse of [Y] one at a time. For more details, see Appendix III.

It may not be obvious that the trapezoidal rule applied to the state variable equations (I.1) leads to the same answers as the trapezoidal rule first applied to individual branch equations, which are then assembled into node equations, as explained in Section 1. The writer has never proved it, but suspects that the answers are identical. For the example of Fig. I.1, this can easily be shown to be true.

The trapezoidal rule of integration is admittedly of lower order accuracy than many other methods, and it is therefore not much discussed in textbooks. It is numerically stable, however, which is usually much more

important in power system transient analysis than accuracy by itself. Numerical stability more or less means that the solution does not "blow up" if Δt is too large; instead, the higher frequencies will be incorrect in the results (in practice, they are usually filtered out), but the lower frequencies for which the chosen Δt provides an appropriate sampling rate will still be reasonably accurate. Fig. I.4 illustrates this for the case of a three-phase line energization. This line was represented as a cascade connection of 18 three-phase nominal π-circuits. The curve for Δt = 5° (based on f = 60 Hz, i.e., Δt = 231.48 μs)

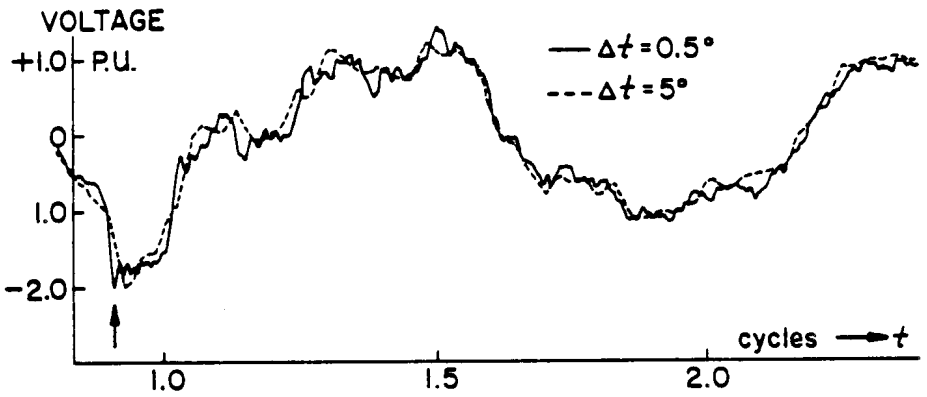

Fig. I.4 - Switching surge overvoltage at the receiving end in a three-phase open-ended line

cannot follow some of the fast oscillations noticeable in the curve for Δt = 0.5°, but the overall accuracy is not too bad. The error between the exact and approximate value at a particular instant in time is obviously not a good measure by itself for overall accuracy, or for the usefulness of a method for these types of studies. In Fig. I.4, an error as large as 0.6 p.u. (at the location of the arrow, assuming that the curve for Δt = 0.5° gives the exact value) is perfectly acceptable, because the overall shape of the overvoltages is still represented with sufficient accuracy.

A physical interpretation of the trapezoidal rule of integration for inductances is given in Section 2.2.1. This interpretation shows that the equations resulting from the trapezoidal rule are identical with the exact solution of a lossless stub-line, for which the answers are always numerically stable though not necessarily as accurate as desired.

I.5 Runge-Kutta Methods

These methods can be used for any system of ordinary differential equations,

$$[\frac{dx}{dt}] = [\, f([x],\ t)\,] \qquad \text{(I.20)}$$

There are many variants of the Runge-Kutta method, but the one most widely used appears to be the following fourth-order method: Starting from the known value [x(t-Δt), the slope is calculated at the point 0 (Fig. I.5(a)),

$$\frac{[\Delta x^{(1)}]}{\Delta t} = [\, f(\, [x(t-\Delta t)\,],\ t-\Delta t\,)\,] \tag{I.21a}$$

which is then used to obtain an approximate value $[x^{(1)}]$ at midpoint 1,

$$[x^{(1)}] = [\, x(t-\Delta t)\,] + \frac{1}{2}\,[\,\Delta x^{(1)}\,] \tag{I.21b}$$

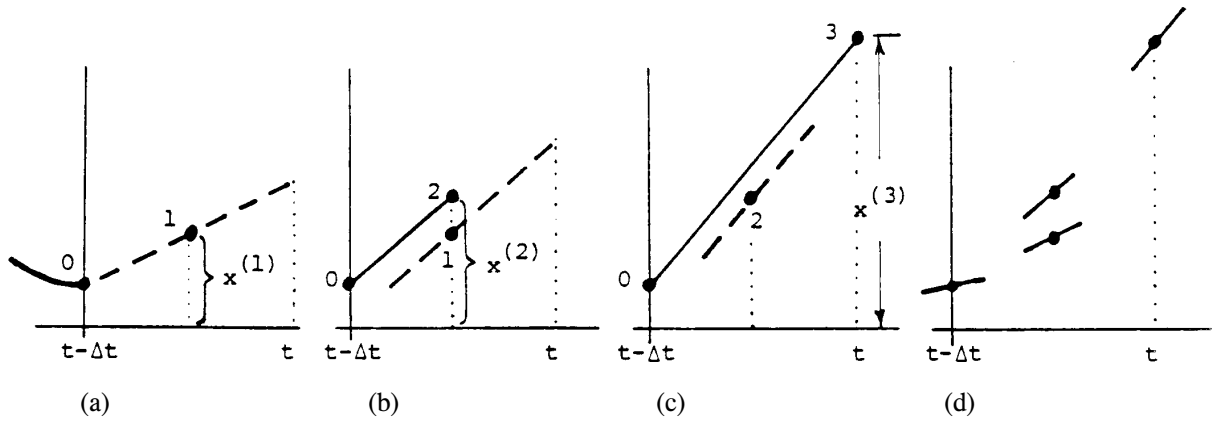

Fig. I.5 - Fourth-order Runge-Kutta method

Now, the slope is recalculated at midpoint 1 (Fig. I.5(b)),

$$\frac{[\,\Delta x^{(2)}\,]}{\Delta t} = [\, f(\, [x^{(1)}],\ t-\Delta\frac{t}{2}\,] \tag{I.21c}$$

and this is used to obtain a second approximate value $[x^{(2)}]$ at midpoint 2,

$$[\,x^{(2)}\,] = [\, x(t-\Delta t)\,] + \frac{1}{2}\,[\,\Delta x^{(2)}\,] \tag{I.21d}$$

Then the slope is evaluated for a third time, now at midpoint 2 (Fig. I.5(c)),

$$\frac{[\,\Delta x^{(3)}\,]}{\Delta t} = [\, f\,(\, [x^{(2)}],\ t-\frac{\Delta t}{2}\,)\,] \tag{I.21e}$$

which is used to get an approximate solution in point 3 at time t,

$$[\,x^{(3)}\,] = [\, x(t-\Delta t)\,] + [\,\Delta x^{(3)}\,] \tag{I.21f}$$

Finally, the slope is evaluated for a fourth time in point 3,

$$\frac{[\,\Delta x^{(4)}\,]}{\Delta t} = [\,f\,(\,[x^{(3)}],\,t\,)\,] \tag{I.21g}$$

From these four slopes in 0, 1, 2, 3 (Fig. I.5(d)), the final value at t is obtained by using their weighted averages,

$$[x(t)] = [x(t-\Delta t)] + \frac{\Delta t}{6} \cdot \left(\frac{[\Delta x^{(1)}]}{\Delta t} + 2\,\frac{[\,\Delta x^{(2)}\,]}{\Delta t} + 2\,\frac{[\,\Delta x^{(3)}\,]}{\Delta t} + \frac{[\,\Delta x^{(4)}\,]}{\Delta t} \right)$$

(I.22)

The mathematical derivation of the Runge-Kutta formula is quite involved (see, for example, in [7]). Intuitively, it can be viewed as an exploration of the "direction field"[4] at a number of sample points (0,1,2,3 in Fig. I.5). There are variants as to the locations of the sample points, and hence as to the weights assigned to them. There are also lower-order Runge-Kutta methods which use fewer sample points.

As already mentioned in Section I.2, the fourth-order Runge-Kutta method of Eq. (I.21) and (I.22) is identical with the fourth-order Taylor series expansion of the transition matrix if the differential equations are linear, at least for autonomous systems with $[g(t)] = 0$ in Eq. (I.1). In that case, Eq. (I.1) becomes

$$\frac{[\,\Delta x^{(1)}\,]}{\Delta t} = [A]\;[x(t-\Delta t)]\;, \qquad [\,x^{(1)}\,] = \left([U] + \frac{\Delta t}{2}[A] \right)\;[x\;(t-\Delta t)]$$

With these values, the second slope becomes

$$\frac{[\Delta x^{(2)}]}{\Delta t} = \left([A] + \frac{\Delta t}{2}\,[A]^2 \right) \cdot [x\;(t-\Delta t)]$$

and

$$[x^{(2)}] = \left([U] + \frac{\Delta t}{2}[A] + \frac{\Delta t^2}{4}[A]^2 \right) \cdot [x(t-\Delta t)]$$

Then the third slope becomes

$$\frac{[\Delta x^{(3)}]}{\Delta t} = \left([A] + \frac{\Delta t}{2}\,[A]^2 + \frac{\Delta t^2}{4}\,[A]^3 \right) \cdot [x(t-\Delta t)]$$

and

$$[x^{(3)}] = \left([U] + \Delta t\;[A] + \frac{\Delta t^2}{2}[A]^2 + \frac{\Delta t^3}{4}[A]^3 \right) \cdot [x(t-\Delta t)]$$

---

[4] If the slopes are calculated at a number of points and graphically displayed as short lines, then one gets a sketch of the "direction field", as indicated in Fig. I.5(d).

from which the fourth slope is calculated as

$$\frac{[\Delta x^{(4)}]}{\Delta t} = \left([A] + \Delta t\,[A]^2 + \frac{\Delta t^2}{2}[A]^3 + \frac{\Delta t^3}{4}[A]^4 \right) \cdot [x(t-\Delta t)]$$

Finally, the new value is obtained with Eq. (I.22) as

$$[x(t)] = \left([U] + \Delta t\,[A] + \frac{\Delta t^2}{2}[A]^2 + \frac{\Delta t^3}{6}[A]^3 + \frac{\Delta t^4}{24}[A]^4 \right) \cdot [x(t-\Delta t)]$$

which is indeed identical with the Taylor series approximation of the transition matrix in Eq. (I.13).

If [A] is zero in Eq. (I.1), that is, if [x] is simply the integral over the known function [g(t)], then the fourth-order Runge-Kutta method is identical with Simpson's rule of integration, in which the curve is approximated as a parabola going through the three known points in t-Δt, t-Δt/2, and t (Fig. I.6).

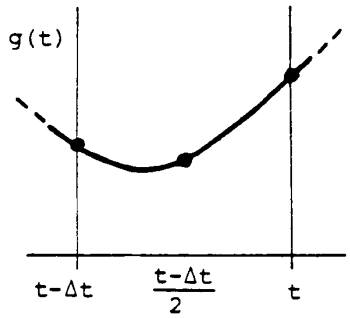

Fig. I.6 - Simpson's rule

The Runge-Kutta method is prone to numerical instability if Δt is not chosen small enough. "It becomes painfully slow in the case of problems having a wide spread of eigenvalues. For the largest eigenvalue (or, equivalently, its reciprocal, the smallest time constant) controls the permissible size of Δt. But the smallest eigenvalues (largest time constants) control the network response and so determine the total length of time over which the integration must be carried out to characterize the response. In the case of a network with a 1000 to 1 ratio of largest to smallest eigenvalue, for instance, it might be necessary to take in the order of 1000 times as many integration steps with the Runge-Kutta method as with some other method which is free of the minimum time-constant barrier" [2}. This problem is indicated in Fig. I.2: Though the ripples may be very small in amplitude, they will cause the slopes to point all over the place, destroying the usefulness of methods based on slopes.

I.6 Predictor-Corrector Methods

These methods can again be used for any system of ordinary differential equations of the type of Eq. (I.20).To explain the basic idea, let us try to apply the trapezoidal rule to Eq. (I.20), which would give us

$$[x^{(h)}] = [x(t-\Delta t)] + \frac{\Delta t}{2} \left([f ([x(t-\Delta t)], t-\Delta t)] + [f ([x^{(h-1)}], t)] \right) \quad \text{(I.23)}$$

In the linear case discussed in Section I.4, this equation could be solved directly for [x]. In the general (time-varying or nonlinear) case, this direct solution is no longer possible, and iterative techniques have to be used. This has already been indicated in Eq. (I.23) by using superscript (h) to indicate the iteration step; at the same time, the argument "t" has been dropped to simplify the notation. The iterative technique works as follows:

1. Use a predictor formula, discussed further on, to obtain a "predicted" guess $[x^{(0)}]$ for the solution at time t.
2. In iteration step h (h=1,2,...), insert the approximate solution $[x^{(h-1)}]$ into the right-hand side of Eq. (I.23) to find a "corrected" solution $[x^{(h)}]$.
3. If the difference between $[x^{(h)}]$ and $[x^{(h-1)}]$ is sufficiently small, then the integration from t-Δt to t is completed. Otherwise, return to step 2.

Eq. (I.23) is a second-order corrector formula. To start the iteration process, a predictor formula is needed for the initial guess $[x^{(0)}]$. A suitable predictor formula for Eq. (I.23) can be obtained from the midpoint rule,

$$[x_{(t)}^{(0)}] = [x(t-2\Delta t)] + 2\Delta t \ [f ([x(t-\Delta t)], t-\Delta t)] \quad \text{(I.24)}$$

or from an extrapolation of known values at t-3Δt, t-2Δt, and t-Δt,

$$[x_{(t)}^{(0)}] = [x(t-3\Delta t)] + \frac{3}{2}\Delta t \left([f([x(t-\Delta t)], t-\Delta t)] + [f([x(t-2\Delta t)], t-2\Delta t)] \right) \quad \text{(I.25)}$$

The difference in step 3 of the iteration scheme gives an estimate of the error, which can be used

(a) to decide whether the step size Δt should be decreased (error too large) or can be increased (error very small), or

(b) to improve the prediction in the next time step.

It is generally better to shorten the step size Δt than to use the corrector formula repeatedly in step 2 above. In using the error estimate to improve the prediction, it is assumed that the difference between the predicted and corrected values changes slowly over successive time steps. This "past experience" can then be used to improve the prediction with a modifier formula. Such a modifier formula for the predictor of Eq. (I.25) and for the corrector of Eq. (I.23) would be

$$[x_{improved}^{(0)}] = [x^{(0)}] + \frac{9}{10} \left([x(t-\Delta t)] - [x_{(t-\Delta t)}^{(0)}] \right) \quad \text{(I.26)}$$

Besides the second-order methods of Eq. (I.23) to Eq. (I.26), there are of course higher-order methods. Fourth-order predictor-corrector methods seem to be used most often. Among these are Milne's method and Hamming's method, with the latter one usually more stable numerically. The theory underlying all predictor-

corrector methods is to pass a polynomial through a number of points at t, t-Δt, t-2Δt, …, and to use this polynomial for integration. The end-point at t is first predicted, and then once or more often corrected. Obviously, the convergence and numerical stability properties of the corrector formula are more important than those of the predictor formula, because the latter is only used to obtain a first guess and determines primarily the number of necessary iteration steps. The predictor and corrector formula should be of the same order in the error terms. There are different classes of predictors: Adams-Bashforth predictors (obtained from integrating Newton' backward interpolation formulas), Milne-type predictors (obtained from an open Newton-Cotes forward-integrating formula), and others. Note that those formulas requiring values at t-2Δt, or further back, are not "self-starting"; Runge-Kutta methods are sometimes used with such formulas to build up enough history points.

It is questionable whether non-self-starting high-order predictor-corrector formulas would be very useful for typical power system transient studies, since waves from distributed-parameter lines hitting lumped elements look almost like discontinuities to the lumped elements, and would therefore require a return to second-order predictor-correctors each time a wave arrives. In linear systems, the second-order corrector of Eq. (I.23) can be solved directly, however, and is then identical with the trapezoidal rule as used in the EMTP.

I.7 Deferred Approach to the Limit (Richardson Extrapolation and Romberg Integration)

The idea behind these methods is fairly simple. Instead of using higher-order methods, the second-order trapezoidal rule (either directly with Eq. (I.17) for linear systems, or iteratively with Eq. (I.23) for more general systems) is used more than once in the interval between t-Δt and t, to improve the accuracy. Assume that the normal step size Δt is used to find $[x^{(1)}]$ at t from [x(t-Δt)], as indicated in Fig. I.7. Now repeat the integration with the trapezoidal rule with half

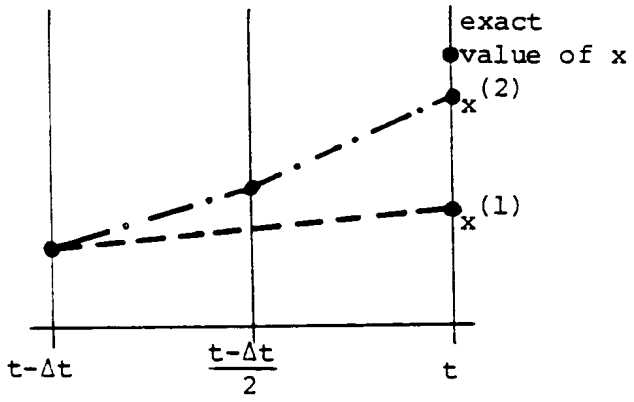

Fig I.7 - Richardson extrapolation

the step size Δt/2, and perform two integration steps to obtain $[x^{(2)}]$. With the two values $[x^{(1)}]$ and $[x^{(2)}]$, an intelligent guess can be made as to where the solution would end up if the step size were decreased more and more. This "extrapolation towards Δt=0" (Richardson's extrapolation) would give us a better answer

$$[x(t)] = [x^{(2)}] + \frac{1}{3}\left([x^{(2)}] - [x^{(1)}] \right) \tag{I.27}$$

The accuracy can be further improved by repeating the integration between t-Δt and t with 4,8,16,… intervals. The corresponding extrapolation formula for Δt → 0 is known as "Romberg integration."

Whether any of these extrapolation formulas are worth the extra computational effort in an EMTP is very difficult to judge. Some numerical analysts seem to feel that these methods look very promising. They offer an elegant accuracy check as well.

I.8 Numerical Stability and Implicit Integration

The writer believes that the numerical stability of the trapezoidal rule has been one of the key factors in making the EMTP such a success. It is therefore worthwhile to expound on this point somewhat more.

The trapezoidal rule belongs to a class of implicit integration schemes, which have recently gained favor amongst numerical mathematicians for the solution of "stiff systems", that is, for systems where the smallest and largest eigenvalues or time constants are orders of magnitude apart [70]. Most power systems are probably stiff in that sense. While implicit integration schemes of higher order than the trapezoidal rule are frequently proposed, their usefulness for the EMTP remains questionable because they are numerically less stable. A fundamental theorem due to Dahlquist [71] states:

Theorem: Let a multistep method be called A-stable, if, when it is applied to the problem [dx/dt] = λ[x], Re(λ) < 0, it is stable for all Δt > 0.

Then: (i) No explicit linear multistep method is A-stable.

(ii) No implicit linear multistep method of order greater than two is A-stable.

(iii) The most accurate A-stable linear multistep method of order two is the trapezoidal rule.

To illustrate the problem of numerical stability, let us assume that a fast oscillation somewhere in the network produces "ripples" of very small amplitudes, which do not have any influence on the overall behavior of the network, similar to those shown in Fig. I.2. Such a mode of oscillation could be described by [72]

$$\frac{d^2x}{dt^2} + x = 0, \quad with\ x(0) = 0, \quad dx/dt(0) = 10^{-4} \tag{I.28}$$

with its exact solution being

$$x = 10^{-4} \sin(t) \tag{I.29}$$

The amplitude of 10^{-4} shall be considered as very small by definition. Eq. (I.28) must be rewritten as a system of first-order differential equations in order to apply any of the numerical solution techniques,

$$\begin{bmatrix} dx_1/dt \\ dx_2/dt \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{I.30}$$

with $x_1 = x$ and $x_2 = dx/dt$. The exact step-by-step solution with Eq. (I.3) is

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = e^{[a]\Delta t} \begin{bmatrix} x_1(t-\Delta t) \\ x_2(t-\Delta t) \end{bmatrix} \tag{I.30a}$$

with

$$[A] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{I.30b}$$

Application of the trapezoidal rule to Eq. (I.30) gives

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \frac{1}{1 + \frac{\Delta t^2}{4}} \begin{bmatrix} 1 - \frac{\Delta t^2}{4} & \Delta t \\ -\Delta t & 1 - \frac{\Delta t^2}{4} \end{bmatrix} \begin{bmatrix} x_1(t-\Delta t) \\ x_2(t-\Delta t) \end{bmatrix} \tag{I.31}$$

It can be shown that

$$x_1^2(t) + x_2^2(t) = x_1^2(t-\Delta t) + x_2^2(t-\Delta t)$$

in Eq. (I.31) for any choice of Δt. Therefore, if the solution is started with the correct initial conditions $x_1^2(0) + x_{22}(0) = 10^{-8}$, the solution for x will always lie between -10^{-4} and $+10^{-4}$, even for step sizes which are much larger than one cycle of oscillation. In other words, the trapezoidal rule "cuts across" oscillations which are very fast but of negligible amplitude, without any danger of numerical instability.

Explicit integration techniques, which include Runge-Kutta methods, are inherently unstable. They require a step size tailored to the highest frequency or smallest time constant (rule of thumb: $\Delta t \leq 0.2\ T_{min}$), even though this mode may produce only negligible ripples, with the overall behavior determined by the larger time constants in stiff systems. Applying the conventional fourth-order Runge-Kutta method to Eq. (I.30) is identical to a fourth-order Taylor series expansion of the transition matrix, as mentioned in Section I.5, and leads to

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 1 - \frac{\Delta t^2}{2} + \frac{\Delta t^4}{24} & \Delta t - \frac{\Delta t^3}{6} \\ -\Delta t + \frac{\Delta t^3}{6} & 1 - \frac{\Delta t^2}{2} + \frac{\Delta t^4}{24} \end{bmatrix} \begin{bmatrix} x_1(t-\Delta t) \\ x_2(t-\Delta t) \end{bmatrix} \tag{I.32}$$

Plotting the curves with a reasonably small Δt, e.g., 6 samples/cycle, reveals that the Runge-Kutta method of Eq. (I.32) is more accurate at first than the trapezoidal rule, but tends to lose the amplitude later on (Fig. I.8). This is not serious since the ripple is assumed to be unimportant in the first place. If the step size is increased, however, to $\Delta t > \sqrt{2}/\pi$ cycles,

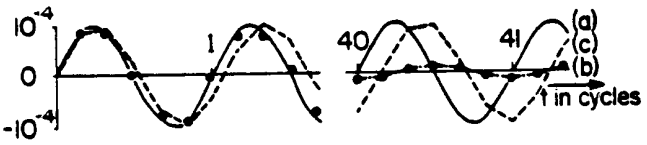

Fig. I.8 - Numerical solution of $d^2x/dt^2 + x = 0$; (a) exact, (b) Runge-Kutta, (c) trapezoidal rule

then the amplitude will eventually grow to infinity. This is illustrated in table I.1 for $\Delta t = 1$ cycle.

Table I.1 - Numerical solution of Eq. (I.28) with $\Delta t = 1$ cycle

t in cycles	1	2	3	4	5	6
exact	0	0	0	0	0	0
trapezoidal rule	$0.58 \cdot 10^{-4}$	$-0.94 \cdot 10^{-4}$	$0.96 \cdot 10^{-4}$	$-0.63 \cdot 10^{-4}$	$0.06 \cdot 10^{-4}$	$0.53 \cdot 10^{-4}$
Runge-Kutta	-0.004	-0.32	-18	-590	-6800	2,600,000

Ref. 72 explains that the trapezoidal rule remains numerically stable even in the limiting case where the time constant T in an equation of the form

$$T\, dx_2/dt = K\, x_1 - x_2 \tag{I.33}$$

becomes zero. For $T = 0$, the trapezoidal rule produces

$$K\, x_1(t) - x_2(t) = -\{ K\, x_1(t-\Delta t) - x_2(t-\Delta t) \} \tag{I.34}$$

which is the correct answer as long as the solution starts from correct initial conditions $K\, x_1(0) - x_2(0) = 0$. Even a slight error in the initial conditions,

$$K\, x_1(0) - x_2(0) = \varepsilon$$

will not cause serious problems. Since Eq. (I.34) just flips the sign of the expression from step to step, the error ε would only produce ripples $\pm\, \varepsilon$ superimposed on the true solution for x_2.

Semlyen and Dabuleanu suggest an implicit third-order integration scheme for the EMTP, in which second-order interpolation (parabola) is used through two known points at $t - 2\,\Delta t$ and $t - \Delta t$, and through the yet unknown solution point at t [73]. Applying this scheme to Eq. (I.30) produces

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x_1(t-\Delta t) \\ x_2(t-\Delta t) \end{bmatrix} + \begin{bmatrix} c & d \\ -d & c \end{bmatrix} \begin{bmatrix} x_1(t-2\Delta t) \\ x_2(t-2\Delta t) \end{bmatrix} \tag{I.35}$$

with

$$a = (1 - \frac{40}{144}\Delta t^2) \ / \ \det$$

$$b = \frac{13}{12}\ \Delta t \ / \ \det$$

$$c = \frac{5}{144}\ \Delta t^2 \ / \ \det$$

$$d = -\frac{\Delta t}{12} \ / \ \det$$

$$\det = 1 + \frac{25}{144}\ \Delta t^2$$

Eq. (I.35) gives indeed higher accuracy than the trapezoidal rule, but only as long as the step size is reasonably small, and as long as the number of steps is not very large. After 40 cycles, with a step size of 6 samples/cycle, Eq. (I.35) would produce peaks which have already grown by a factor of 20,000. This indicates that the choice of the step in Eq. (I.35) is subject to limitations imposed by numerical stability considerations, whereas the trapezoidal rule is not. A step size of 6 samples/cycles is not too large for fast oscillations which have no influence on the overall behavior. The trapezoidal rule simply filters them out. High-order implicit integration schemes are therefore not as useful for the EMTP as one might be thought to believe from recent literature on implicit integration schemes for stiff systems.

I.9 Backward Euler Method

The major drawback of the trapezoidal rule of integration of Section I.4 is the danger of numerical oscillations when it is used as a differentiator, e.g., in

$$v = L\ di / dt \tag{I.36}$$

with current i being the forcing function. A sudden jump in di/dt, which could be caused by current interruption in a circuit breaker, should create a sudden jump in the voltage v. Instead, the trapezoidal rule of integration produces undamped numerical oscillations around the correct answer, as explained in Section 2.2.2. These oscillations can be damped out by adding a parallel resistor R_p across the inductance. Section 2.2.2 shows that critical damping is achieved if $R_p = 2L/\Delta t$. In that case, the "damped" trapezoidal rule of Eq. (2.20) transforms Eq. (I.36) into

$$v(t) = \frac{L}{\Delta t}\ [i(t) - i(t-\Delta t)] \tag{I.37}$$

which is simply the backward Euler method. Therefore, the "critically damped" trapezoidal rule and the backward Euler method are identical.

In general, the undamped trapezoidal rule is better than the backward Euler method, because the latter method produces too much damping. It is a good method, however, if it is only used for a few steps to get over instants of discontinuities (see Appendix II).