SOME EXACT SOLUTIONS TO APPROXIMATE LINEAR AND NON-LINEAR VIBRATION PROBLEMS

by

THAD ALAN KING

B. S., Kansas State University, 1961

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1966

Approved by:

F. C. Appl
Major Professor

TABLE OF CONTENTS

NOMENCLATURE

A	coefficients of solution
B_1, B_2	parameters of spring coefficient
c	damper coefficient
C	ratio of damper coefficient to mass
F	spring force
k	spring coefficient
K	ratio of spring coefficient to mass
i, j, m, n	integers
m	mass
x, z	displacement of mass from equilibrium position

INTRODUCTION

For many vibrating systems, displacement information may be obtained by simplifications which reduce the system to a linear spring-mass-damper arrangement with constant spring and damper coefficients. The solution of the resulting differential equation of motion of the mass is easily obtained. Although the solution is exact, it is still an approximate solution for the original system. The degree of approximation, of course, depends upon the similarity of the two systems. A solution determined in this manner for a highly non-linear system may be inadequate.

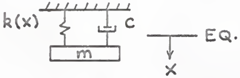
There is a need for additional techniques to handle the non-linear cases. Through recent studies involving other types of problems by Appl and Zorowski [1], Appl and Byers[2], and Appl and Hung[3], such a technique has been found. It involves the idea of determining an exact solution to an approximate problem. In some types of problems upper and lower bounds for the initial problem can be established in terms of the approximate problem solutions. If the problem is bounded, the degree of approximation is known.

This report is the initial step in attempting to apply this technique to a vibration problem. It is an investigation into a method for obtaining exact solutions for approximate problems. Two methods are applied, steepest descent, and the method of Runge-Kutta.

SOME EXACT SOLUTIONS TO APPROXIMATE LINEAR AND NON-LINEAR VIBRATION PROBLEMS

Formulation of Problem and Derivation of Equations

Consider a free vibration system composed of a mass, damper, and spring, where the spring coefficient, k, is a non-linear function of the displacement, x, measured from the equilibrium position.



The differential equation of motion of the mass is

$$\ddot{x} + C\dot{x} + K(x)x = 0 \qquad (1)$$

where

$$C = \frac{c}{m}$$

$$K(x) = \frac{k(x)}{m}$$

For this investigation, let

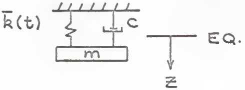
$$K(x) = B_1 + B_2 x^2$$

where B_1 and B_2 are constants. Note that for $B_2 = 0$, the system becomes linear.

The force exerted by the spring on the mass is

$$F(x) = -B_1 x - B_2 x^3 \qquad (2)$$

At this point consider another vibration system identical to the previous one except that now the spring coefficient is a function of time, t, instead of the displacement, z.



The differential equation for this case is

$$\ddot{z} + C\dot{z} + \bar{K}(t) z = 0$$

with a spring force

$$\bar{F}(t) = -\bar{K}(t) z = \ddot{z} + C\dot{z} \qquad (3)$$

The two systems now differ only by the spring forces. Suppose that a solution z(t) is assumed in the form

$$z(t) = A_0 + A_1 t + A_2 t^2 + A_3 t^3 + \ldots + A_n t^n$$

Then,

$$\dot{z}(t) = A_1 + 2A_2t + 3A_3t^2 + \ldots + (n)A_nt^{(n-1)}$$

and

$$\ddot{z}(t) = 2A_2 + 6A_3t + 12A_4t^2 + \ldots + (n)(n-1)A_nt^{(n-2)}$$

These relations and (3) determine $\overline{F}(t)$. Thus, $z(t)$ is the exact solution for differential equation (3) which can be written

$$\ddot{z} + C\dot{z} - \overline{F}(t) = 0$$

If $\overline{F}(t) \equiv F[z(t)]$ it means that the spring force in the second system is the same as for the first case. Thus, the two systems are idential in every respect so that

$$x = z = A_0 + A_1t + A_2t^2 + \ldots + A_nt^n$$

is the exact solution for the initial vibrating problem.

For $\overline{F}(t) \simeq F[z(t)]$, $z(t)$ is the exact solution for the approximate problem described by

$$\ddot{z} + C\dot{z} + \overline{K}(t)z = 0$$

Thus an exact solution is known for an approximate problem. The degree of approximation is unknown and is beyond the scope of this report. However, some indication of the accuracy is given by determining how nearly $\tilde{F}(t)$ equals $F\left[z(t)\right]$. This, in turn, depends upon the coefficients in the solution $z(t)$.

Application of the Method of Runge-Kutta

Consider the differential equation

$$\ddot{z} + C\dot{z} + K(z)z = 0 \qquad (4)$$

with the initial conditions

$$t = 0; \quad z = z_0$$
$$\dot{z} = 0$$

Introducing the transformation

$$\dot{Y} = \ddot{z}$$
$$Y = \dot{z}$$

Equation (4) can be written

$$\dot{Y} + CY + K(z)z = 0$$

This, together with the transformation, yields two simultaneous first-order differential equations suitable for application of the method of Runge-Kutta [4].

After a period of time, Δt, has elapsed, the values of z, y, and t, become

$$z_1 = z_0 + \Delta z$$

$$Y_1 = Y_0 + \Delta Y$$

$$t_1 = t_0 + \Delta t$$

The changes, Δz and Δy, are approximated as follows:

$$\dot{y} = f_1(t,z,y)$$

$$\dot{z} = f_2(t,z,y)$$

$$\Delta y = \frac{1}{6}(s_1 + 2s_2 + 2s_3 + s_4)$$

$$\Delta z = \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4)$$

where,

$$s_1 = f_1(t_0, z_0, y_0)\Delta t$$

$$s_2 = f_1\left(t_0 + \frac{\Delta t}{2}, z_0 + \frac{l_1}{2}, y_0 + \frac{s_1}{2}\right)\Delta t$$

$$s_3 = f_1\left(t_0 + \frac{\Delta t}{2}, z_0 + \frac{l_2}{2}, y_0 + \frac{s_2}{2}\right)\Delta t$$

$$s_4 = f_1(t_0 + \Delta t, z_0 + l_3, y_0 + s_3)\Delta t$$

and

$$l_1 = f_2(t_0, z_0, y_0)\Delta t$$

$$l_2 = f_2\left(t_0 + \frac{\Delta t}{2}, z_0 + \frac{l_1}{2}, y_0 + \frac{s_1}{2}\right)\Delta t$$

$$l_3 = f_2\left(t_0 + \frac{\Delta t}{2}, z_0 + \frac{l_2}{2}, y_0 + \frac{s_2}{2}\right)\Delta t$$

$$l_4 = f_2(t_0 + \Delta t, z_0 + l_3, y_0 + s_3)\Delta t$$

Once z_1 and y_1 are determined at time t_1, another Δt is chosen. Using z_1 and y_1 as "initial conditions," the procedure is repeated to approximate z_2 and y_2 for time t_2. Additional iterations are effected until the desired displacements and velocities, z and y, are known at specific times over the period involved. This procedure can be easily programmed for execution on a digital computer. The program used is shown in Appendix B.

By assuming the polynomial form for z(t), a linear algebraic equation can be written for each z and its corresponding time value. The resulting set of equations can be solved to determine the coefficients, A_1, A_2........, A_n. Once these are known, z(t) is established in polynomial form.

Application of the Method of Steepest Descent

This method [3] is employed to improve the function z(t) by determining a more suitable set of coefficients in the solution.

A deviation function, α , is created from residual terms given by

$$R(t_j) = \bar{F}(t_j) - F[z(t_j)]$$

where t_j denotes particular time values. Then,

$$\alpha = \sum_{j=1}^{m} \left[R(t_j) \right]^2$$

The function z(t) is improved as α is reduced.

The function α may be considered as a function of $\bar{A}$ in n-dimensional space where first and second derivatives exist. Vector $\bar{A}$ is considered to be the displacement vector,

$$\bar{A} = (A_1, A_2, \ldots, A_i, \ldots, A_n)$$

The method of steepest descent is used to minimize α by improving an assumed $\bar{A}$. This is achieved in this case by changing each coefficient through an iterative scheme. The initial $\bar{A}$ is comprised of the coefficients determined through the Runge-Kutta method.

Basically, the change for each A_i is

$$\Delta A_i = -\frac{\alpha}{\gamma(\nabla\alpha \cdot \nabla\alpha)} \times \frac{\partial \alpha}{\partial A_i}$$

where γ is a necessary constant which allows control of the amount of change.

For this problem,

$$\frac{\partial \alpha}{\partial A_i} = 2 \sum_{j=1}^{m} R(t_j) \frac{\partial}{\partial A_i} \left\{ R(t_j) \right\}$$

$$\nabla \alpha = \left\{ \frac{\partial \alpha}{\partial A_1}, \frac{\partial \alpha}{\partial A_2}, \ldots, \frac{\partial \alpha}{\partial A_i}, \ldots, \frac{\partial \alpha}{\partial A_n} \right\}$$

After all the ΔA_i's have been determined for a given set of coefficients, $A_{i,1}$, the change

$$A_{i,2} = A_{i,1} + \Delta A_{i,1}$$

is made and the new set, $A_{i,2}$, is then improved in the same manner. The process is repeated until α has been reduced to the desired value. Appendix C contains the basic fortran program used.

EXAMPLE PROBLEMS

Three systems were chosen for investigation. All cases are for a mass-damper-spring arrangement which is initially displaced a distance $z_0 = 1.0$ from the equilibrium position and released with zero velocity.

The method for obtaining a polynomial solution from the method of Runge-Kutta is identical for the three problems. The time interval over which the function is investigated is from $t = 0.0$ to $t = 1.0$ with a time increment, $\Delta t = 0.01$. Twenty algebraic equations are solved simultaneously giving twenty coefficients. Three additional coefficients are determined exactly by applying the initial conditions to the differential equation. They are,

$$A_1 = 1.0$$

$$A_2 = 0.0$$

$$A_3 = -\tfrac{1}{2}(B_1 + B_2)$$

The solution $z(t)$ as obtained from Runge-Kutta, therefore, consists of twenty-three terms.

Steepest descent is also employed in all cases. Again, the first three coefficients retained their exact values. For Case 1 and Case 3, the remaining twenty coefficients from the solution of the algebraic equations form the initial displacement vector. Residuals, $R(t_j)$, are determined for various values of t. The sum of the squares of these residuals is, then, the deviation function α.

Case 1. Linear undamped system. For this problem, the values of the parameters are

$$C = 0.0$$

$$B_1 = 80.0$$

$$B_2 = 0.0$$

$$\therefore K(x) = 80.0 = \text{constant}$$

The differential equation for the linear case can be solved exactly yielding

$$z(t) = z_0 \cos\sqrt{K}\, t$$

This gives a means of checking the z(t) polynomial and the discrete values of the displacement as obtained by applying Runge-Kutta.

Case 2. Non-linear damped system. Here,

$$C = 1.0$$

$$B_1 = 80.0$$

$$B_2 = 10.0$$

$$\therefore K(x) = 80.0 + 10.0x^2$$

Case 3. Non-linear damped system. The values of the parameters are

$$C = 1.0$$

$$B_1 = 40.0$$

$$B_2 = 40.0$$

$$\therefore K(x) = 40.0 + 40.0x^2$$

The results for each case are presented in tabular form, listing values determined for z for numerous time values. Table 1 lists the percent error existing between the exact displacement for the linear case and the numerical results obtained for both the discrete values of z directly from the method of Runge-Kutta and the functional values determined from the resulting polynomial.

Tables 2, 4 and 6 give the values of the polynomial as determined from Runge-Kutta for Cases 1, 2 and 3, respectively. The degree of approximation between the actual problem and the approximate problem, for which z is the exact solution, is indicated by the spring force percent error.

$$\% \text{ Error} = \frac{\bar{F}(t) - F[z(t)]}{F[z(t)]} \times 100.0$$

The effect of the method of steepest descent when applied to the coefficients obtained through Runge-Kutta is demonstrated in Tables 3, 5 and 7. Again the spring force percent error is utilized. Initial and final values of the deviation function, α, are shown at the end of each table.

Appendix A contains graphs of the z(t) polynomial from the method of Runge-Kutta for each of the three problems.

Table 1. Comparison of exact displacements with numerical values obtained directly from Runge-Kutta and functional values from the polynomial for Runge-Kutta for Case 1.
$C = 0.0,\ B_1 = 80.0,\ B_2 = 0.0,\ \Delta t = 0.01$

Time	z Exact	Numerical values directly from Runge-Kutta: z	Numerical values directly from Runge-Kutta: % Error	Functional values from polynomial for Runge-Kutta: z	Functional values from polynomial for Runge-Kutta: % Error
.02	.984042621181	.984042631118	.00000101	.984042631180	.00000102
.12	.477215760016	.477216241258	.00010085	.477216241223	.00010084
.22	-.386601298672	-.386600303363	-.00025745	-.386600303370	-.00025745
.32	-.961214006752	-.961213477687	-.00005504	-.961213477688	-.00005504
.42	-.816772553416	-.816773584213	.00012620	-.816773584213	.00012620
.52	-.061329076245	-.061331533722	.00400703	-.0613315337228	.00400707
.62	.739992766388	.739990619795	-.00029008	.739990619794	-.00029008
.72	.987749144567	.987749426389	.00002853	.987749426382	.00002853
.82	.496601253928	.496604494796	.00065261	.496604494723	.00065259
.92	-.366038516451	-.366034324414	-.00114524	-.366034327222	-.00114447
.98	-.790365173580	-.790362062633	-.00039613	-.790361976989	-.00040444

Table 2. Values of z(t) polynomial obtained from Runge-Kutta method for Case 1. C = 0.0, B_1 = 80.0 B_2 = 0.0, Number of coefficients = 23

Time	z	$\overline{F}(t)$	F[z(t)]	% Error
.000	1.00000	-80.00000	-80.00000	.00000
.075	.78331	-62.66490	-62.66497	-.00011
.150	.22716	-18.17245	-18.17247	-.00014
.225	-.42744	34.19550	34.19553	-.00009
.300	-.89680	71.74395	71.74402	-.00010
.375	-.97750	78.20031	78.20040	-.00011
.450	-.63458	50.76657	50.76663	-.00012
.525	-.01665	1.33183	1.33184	-.00058
.600	.60850	-48.68009	-48.68014	-.00010
.675	.96994	-77.59524	-77.59532	-.00010
.750	.91103	-72.88250	-72.88258	-.00011
.825	.45730	-36.58426	-36.58430	-.00012
.900	-.19461	15.56870	15.56871	-.00011
.975	-.76218	60.97457	60.97463	-.00009
1.000	-.88676	70.93819	70.94069	-.00352

Table 3. Effect of Steepest Descent Applied to Polynomial from Runge-Kutta for Case 1. C = 0.0, B_1 = 80.0 B_2 = 0.0, Number of coefficients = 23

	Displacement from Polynomial for Runge-Kutta		Displacement from Polynomial for Steepest Descent	
Time	z	Spring Force % Error	z	Spring Force % Error
.050	.90165568	-.000109	.90165568	-.000108
.100	.62596597	-.000117	.62596597	-.000115
.150	.22715590	-.000140	.22715590	-.000131
.200	-.21633313	-.000070	-.21633314	-.000087
.250	-.61727189	-.000096	-.61727191	-.000105
.300	-.89680028	-.000102	-.89680032	-.000112
.350	-.99993828	-.000106	-.99993835	-.000120
.400	-.90639981	-.000110	-.906399921	-.000132
.450	-.63458284	-.000116	-.63458300	-.000161
.500	-.23795066	-.000139	-.23795091	-.000311
.550	.20548368	-.000068	.20548332	.000213
.600	.60850171	-.000096	.60850119	.000038
.650	.89183438	-.000102	.89183364	.000024
.700	.99975338	-.000106	.99975234	.000049
.750	.91103228	-.000110	.91103085	.000115
.800	.64312152	-.000115	.64311957	.000274
.850	.24871610	-.000141	.24871353	.000844
.900	-.19460893	-.000108	-.19461207	-.000385
.950	-.59965658	.000075	-.59965969	.001208
1.000	-.88675861	-.003525	-.88675952	-.000328
	α = .000006360		α = .0000005004	

Table 4. Values of z(t) polynomial obtained from Runge-Kutta method for Case 2. $C = 1.0$, $B_1 = 80.0$ $B_2 = 10.0$, Number of coefficients = 123

Time	z	$\overline{F}(t)$	F[z(t)]	% Error
.000	1.00000	-90.00000	-90.00000	.00000
.075	.76520	-65.73663	-65.69633	.06133
.150	.20372	-16.38849	-16.38248	.03670
.225	-.40455	33.02932	33.02626	.00927
.300	-.79423	68.54896	68.54864	.00046
.375	-.80008	69.12765	69.12841	-.00109
.450	-.44702	36.65491	36.65498	-.00018
.525	.07367	- 5.89694	- 5.89732	-.00656
.600	.52553	-43.49337	-43.49339	-.00007
.675	.71712	-61.05769	-61.05734	.00057
.750	.57830	-48.19924	-48.19830	.00194
.825	.19799	-15.92447	-15.91686	.04782
.900	-.23878	19.66940	19.23867	2.23885
.975	-.54662	43.53450	45.36338	-4.03172
1.000	-.59012	247.58990	49.26425	402.57523

Table 5. Effect of Steepest Descent Applied to Polynomial from Runge-Kutta for Case 2. C = 1.0, B_1 = 80.0 B_2 = 10.0, Number of coefficients = 28

	Displacement from Polynomial for Runge-Kutta		Displacement from Polynomial for Steepest Descent	
Time	z	Spring Force % Error	z	Spring Force % Error
.04	.92997599	-.00074	.92997687	-.00511
.12	.45026398	-.00089	.45028993	-.04227
.20	-.21413690	.00013	-.21400462	.22196
.28	-.72384582	-.00016	-.72344200	.14291
.36	-.83148979	-.00023	-.83052274	.25213
.44	-.50893267	-.00058	-.50691102	.76512
.52	.03856511	-.00406	.04246568	-16.41699
.60	.52552518	-.00074	.53271980	-2.42934
.68	.71797696	-.00038	.73099089	-3.27714
.76	.53804239	-.00315	.56148598	-7.43685
.84	.10868924	-.16130	.15001006	-37.20801
.92	-.33952395	.90404	-.27953439	-22.55675
1.00	-.58935300	236.64735	-.61279742	21.32421
	α = 13,561.4		α = 1,402.7	

Table 6. Values of z(t) polynomial obtained from Runge-Kutta method for Case 3. C = 1.0, B_1 = 40.0 B_2 = 40.0, Number of coefficients = 23

Time	z	$\overline{F}(t)$	F[z(t)]	% Error
.000	1.00000	-80.00000	-80.00000	.00000
.075	.79542	-51.77067	-51.94659	-.33866
.150	.32826	-14.51888	-14.54512	-.18041
.225	-.19091	7.90473	7.91487	-.12813
.300	-.62601	34.85169	34.85346	-.00508
.375	-.84096	57.42775	57.42772	.00004
.450	-.74579	46.42432	46.42378	.00116
.525	-.41389	19.39414	19.39142	.01398
.600	.00261	- .10705	- .10456	2.38072
.675	.38849	-17.90136	-17.88523	.09021
.750	.64701	-36.73012	-36.78666	-.15370
.825	.69528	-40.86585	-41.25511	-.94355
.900	.52535	-37.08918	-26.81360	38.32224
.975	.30345	.33828	-13.25548	-102.55237
1.000	.11225	-3821.90872	- 4.54641	83,964.33809

Table 7. Effect of Steepest Descent Applied to Polynomial from Runge-Kutta for Case 3. $C = 1.0$, $B_1 = 40.0$ $B_2 = 40.0$, Number of coefficients = 23

	Displacement from Polynomial for Runge-Kutta		Displacement from Polynomial for Steepest Descent	
Time	z	Spring Force % Error	z	Spring Force % Error
.050	.9048	-.1290	.9048	-.1043
.100	.6568	.1907	.6568	.2942
.150	.3283	-.1804	.3281	.2768
.200	-.02122	-1.125	-.0216	-12.87
.250	-.3515	.0244	-.3524	-1.021
.300	-.6260	-.0051	-.6276	-.8454
.350	-.8021	.0016	-.8048	-.9548
.400	-.8436	-.0013	-.8478	-1.294
.450	-.7458	.0011	-.7519	-1.933
.500	-.5411	-.0026	-.5497	-3.071
.550	-.2778	.0101	-.2893	-5.180
.600	.0026	2.418	-.0118	-40.02
.650	.2691	-.0491	.2531	-8.348
.700	.4939	.0692	.4808	-17.15
.750	.6470	-.1550	.6507	-37.00
.800	.7058	.5603	.7593	-83.66
.850	.6599	-3.484	.8527	-186.8
.900	.5253	38.35	1.080	-292.6
.950	.3322	-881.4	1.800	-3109.
1.00	.1123	83,960.	3.824	7288.
	α = 14,589,000		α = 1,020,000	

RESULTS

The numerical values for the displacement given by Runge-Kutta in the linear problem are very accurate, as shown in Table 1, with a maximum error of .0040%. This error can be reduced if desired by reducing the time increment, Δt, but for this case the results seemed adequate.

For a similar linear problem ($B_1 \simeq 40.0$) a test was made using $\Delta t = .025$. Values for z were determined through twenty periods of vibration. At the end cycle ($t = 20$), the results were compared with the exact solution and the maximum percent error was calculated to be .094%.

The polynomial determined from the numerical values for z is also reasonably accurate for the linear case. As shown by Table 1, the percent error existing is almost identical to the error resulting from the method of Runge-Kutta. This accuracy is reflected in Table 2 where it is shown that $F(t)$ and $F[z(t)]$ differ by a maximum of only .0035% occuring at $t = 1.0$. This, in turn, means the approximate linear problem is a very good approximation for the actual problem.

For the non-linear cases, 2 and 3, a somewhat different situation exists. The values of the polynomial solution appear adequate except near the end of the time interval. Tables 4 and 6 show reasonably small spring force errors for $0 \leq t \leq 0.90$. However, for $t = 1.0$, the percent error becomes 402.6% and 83,964.3% for Cases 2 and 3 respectively. A corresponding value for Case 2 using twenty-eight coefficients instead of the twenty-three was 235.4%.

Utilization of the method of steepest descent had a marked effect on all cases. While the spring force percent error increased for some values of t and decreased for others, the deviation function α was reduced. This is best illustrated by Case 2 where as an example, for t = 0.6, the percent error increased from -.00007% to -2.42% as a result of the application of the method of steepest descent. However, α , the sum of the residuals squared was reduced from 13,600 to 1,400. It may be further reduced by additional iterations.

It was noted that the γ factor used in the method of steepest descent was rather critical. For $\gamma = 1.0$, α decreased for several iterations but suddenly increased. Further iterations continued the reduction until another increase occurred. By increasing the value of γ , α continued decreasing to a smaller value before an increase was noted. The amount of decrease in α , however, appeared smaller for each iteration.

CONCLUSION

The results of the example problems indicate that the method outlined will provide exact solutions to approximate problems. The polynomial from the method of Runge-Kutta provides good results for the linear case and further improvement is made by the application of steepest descent.

The polynomial from Runge-Kutta is increasingly less accurate as the system becomes more and more non-linear. In this case, it is necessary to apply the method of steepest descent to improve the results to an adequate level. However, the computer time required for steepest descent yields this procedure undesirable for highly non-linear problems.

A major factor seems to be the number of coefficients in the solution. Better results for a non-linear system are obtained by increasing this number from twenty-three to twenty-eight, the maximum which was obtained with the 1410 computer.

It is, therefore, concluded that the use of a larger and faster computer is essential for the success of this procedure when applied to non-linear problems.

REFERENCES

1. F. C. Appl and C. F. Zorowski, "Upper and Lower Bounds for Special Eigenvalues," Journal of Applied Mechanics, Vol. 26, No. 2, 1959.

2. F. C. Appl and N. R. Byers, "Fundamental Frequency of Simple Supported Rectangular Plates with Linearly Varying Thickness," Journal of Applied Mechanics, No. 64, March 1965.

3. F. C. Appl and H. M. Hung, "A Principle of Convergent Bounding Analysis with Application to Heat Transfer through Fins," International Journal of Mechanical Sciences, Vol. 6, 1964.

4. J. B. Scarborough, Numerical Mathematic Analysis, New York, New York,: Johns Hopkins Press, 1950.

APPENDIX A

Displacement Curves from
Polynomial for Runge-Kutta

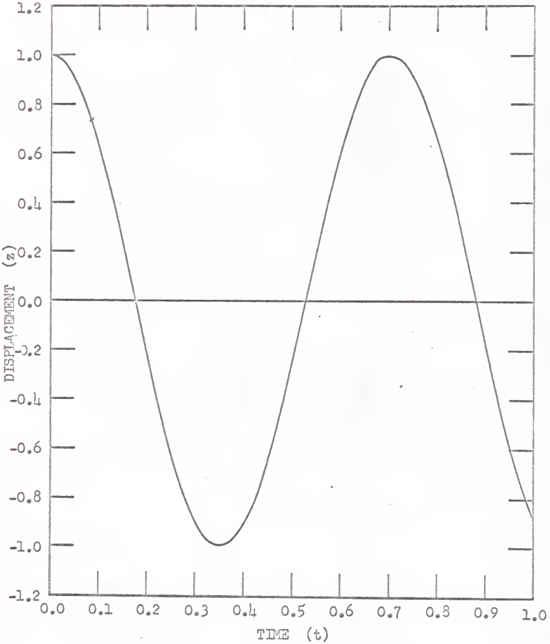


Figure 1. z(t) Curve for Case 1 from Polynomial for Runge-Kutta. $C = 0.0$, $B_1 = 80.0$, $B_2 = 0.0$, $0 \leq t \leq 1.0$.

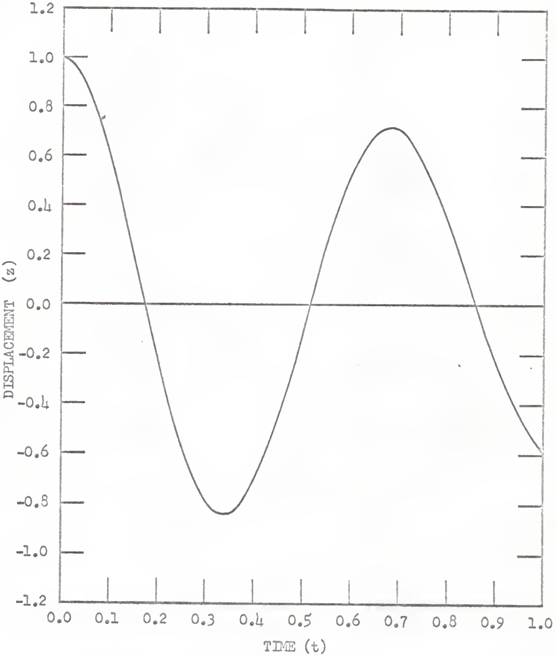


Figure 2. Displacement vs. Time Curve for Polynomial z(t) Obtained from Runge-Kutta for Case 2. $C = 1.0$, $B_1 = 80.0$, $B_2 = 10.0$

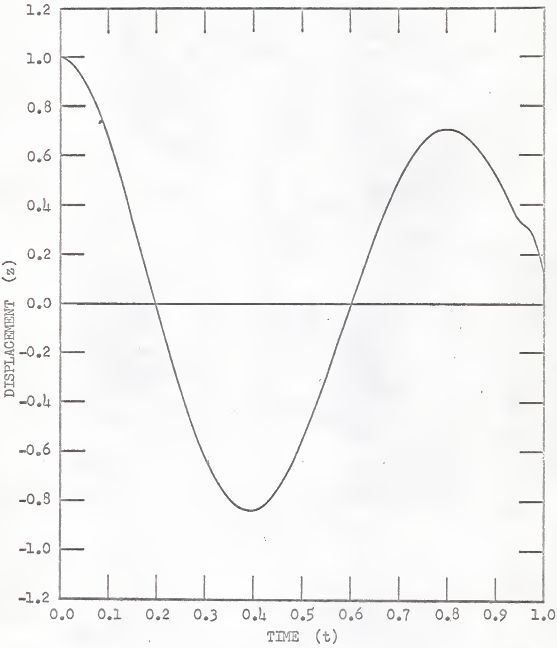


Figure 3. Displacement vs. Time Curve for Polynomial z(t) Obtained from Runge-Kutta for Case 3. $C = 1.0$, $B_1 = 40.0$, $B_2 = 40.0$

APPENDIX B

Fortran Program for
Runge-Kutta Method

BASIC RUNGE-KUTTA PROGRAM

```
 10 FORMAT(E16.8)
 11 FORMAT(I5)
 12 FORMAT(4E16.8)
 14 FORMAT(8X,4HTIME,7X,14HX DISPLACEMENT,4X,10HZ VELOCITY)
 16 FORMAT(6E16.8,I10)
 17 FORMAT(1H )
    READ(1,10)C
    READ(1,10)B1
    READ(1,10)B2
    READ(1,10)X              (INITIAL DISPLACEMENT)
    READ(1,10)Z              (INITIAL VELOCITY)
    READ(1,10)DELT           (DELTA T)
    READ(1,11)NINT           (NUMBER OF STEPS)
    WRITE(3,15)
    WRITE(3,16)C,B1,B2,DELT,NINT
    WRITE(3,17)
    WRITE(3,14)
    T=0.0                    (INITIAL TIME)
    WRITE(3,12)T,X,Z
    DO100N=1,NINT
    SK1=-C*Z-B1*X-B2*X*X*X)*DELT
    SL1=Z*DELT
    T2=T+(DELT/2.0)
    Z1=Z+(SK1/2.0)
    X1=X+(SL1/2.0)
    SK2=-C*Z1-B1*X1-B2*X1*X1*X1)*DELT
    SL2=Z1*DELT
    Z2=Z+(SK2/2.0)
    X2=X+(SL2/2.0)
    SK3=-C*Z2-B1*X2-B2*X2*X2*X2)*DELT
    SL3=Z2*DELT
    Z3=Z+SK3
    X3=X+SL3
    T=T+DELT
    SK4=C*Z3-B1*X3-B2*X3*X3*X3)*DELT
    SL4=Z3*DELT
    DELX=(SL1+2.0*SL2+2.0*SL3+SL4)/6.0   (CHANGE IN DISPLACEMENT X)
    DELZ=(SK1+2.0*SK2+2.0*SK3+SK4)/6.0   (CHANGE IN VELOCITY Z)
    X=X+DELX                 (CHANGE X TO BEGIN NEXT TIME INCREMENT)
    Z=Z+DELZ                 (CHANGE Z TO BEGIN NEXT TIME INCREMENT)
    WRITE(3,12)T,X,Z
    NQ=(N/10)*10
    IF(N.EQ.NQ)WRITE(2,12)T,X,Z
100 CONTINUE                 (BEGIN NEXT STEP)
    STOP
    END
```

APPENDIX C

Fortran Program for
Steepest Descent Method

BASIC STEEPEST DESCENT PROGRAM

```
      DIMENSION A(29),T(28),PX(28),PXD(28),PXDD(28),GALP(28)
   10 FORMAT(E16.8)
   11 FORMAT(I5)
   13 FORMAT(7X,1HC,15X,2HB1,14X,2HB2,9X,5HNCOEF,3X,5HNTIME,3X,
     25HGAMMA,12X,3HTAU,12X,5HTZERO,9X,5HNITER)
   14 FORMAT(3E16.8,2I6,3E16.8,I7)
   17 FORMAT(I20,E24.16)
   18 FORMAT(19X,1HI,9X,8HOLD A(I))
   19 FORMAT(E24.16)
   20 FORMAT(19X,1HI,9X,8HNEW A(I))
   26 FORMAT(E16.8,4E24.16)
   27 FORMAT(1X,3HTRY)
   28 FORMAT(I3)
   30 FORMAT(7X,4HTIME,12X,14HX DISPLACEMENT,13X,5HF BAR,15X,
     212HSPRING FORCE,13X,13HPERCENT ERROR)
   40 FORMAT(15X,6HALPHA=,E24.16)
      READ(1,11) NCOEF
      READ(1,11) NTIME
      READ(1,11) NITER
      READ(1,19)(A(I),I=1,NCOEF)  (COEFFICIENTS FROM R.K. POLY)
      READ(1,10) C
      READ(1,10) B1
      READ(1,10) B2
      READ(1,10) GAMMA
      READ(1,10) TAU
      READ(1,10) TZERO
      WRITE(3,18)
      WRITE(3,17)(I,A(I),I=1,NCOEF)
      WRITE(3,13)
      WRITE(3,14)C,B1,B2,NCOEF,NTIME,GAMMA,TAU,TZERO,NITER
      A(3)=-(B1*A(1)+B2*A(1)*A(1)*A(1))/2.0
      DELTAU=TAU/(FLOAT(NTIME-1)) (DETERMINE T(J))
      NTRY=NITER
      DO150N=1,NITER              (BEGIN ITERATION)
      TT=TZERO
      IF(N.LT.2) GO TO 205
      IF(N.LT.(NTRY-5)) GO TO 33
  205 WRITE(3,27)
      WRITE(3,28)N
      WRITE(3,30)
   33 ALP=0.0
      DO109I=4,NCOEF
  109 GALP(I)=0.0
      SGALP=0.0
      DO100J=1,NTIME
      T(1)=1.0
      DO110K=2,NCOEF
  110 T(K)=T(K-1)*TT
      PXD(1)=0.0
      PXDD(1)=0.0
      PXDD(2)=0.0
      X=0.0
```

BASIC STEEPEST DESCENT PROGRAM (CONTINUED)

```
    DO 50I=1,NCOEF
    PX(I)=T(I)
    K=I+1
    IF(K.GT.NCOEF) GO TO 48
    PXD(K)=FLOAT(I)*T(I)
    K=I+2
    IF(K.GT.NCOEF) GO TO 48
    ZA1=FLOAT(I)
    ZA2=ZA1+1.0
 48 PXDD(K)=ZA1*ZA2*T(I)
 50 X=X+A(I)*T(I)
    XD=0.0
    DO170I=2,NCOEF
    XYZ=I-1
170 XD=XD+XYZ*A(I)*T(I-1)
    XDD=0.0
    DO180I=3,NCOEF
    XYZ=I-1
180 XDD=XDD+XYZ*(XYZ-1.0)*A(I)*T(I-2)
    AKT=XDD+C*XD                   (F BAR (T))
    AKNC=-B1*X-B2*X*X*X            (F(Z(T)))
    RB=AKT-AKNC                    (RESIDUAL)
    PER=RB*100.0/AKNC              (PERCENT ERROR)
    IF(N.LT.2) GO TO 215
    IF(N.LT.(NTRY-5)) GO TO 216
215 WRITE(3,26)TT,X,AKT,AKNC,PER
216 ALP=ALP+RB*RB                  (CALCULATE ALPHA)
    DO200I=4,NCOEF
    PR=PXDD(I)+C*PXD(I)+B1*PX(I)+3.0*B2*X*X*PX(I)
200 GALP(I)=GALP(I)+2.0*RB*PR      (CALCULATE GRADIENT ALPHA)
100 TT=TT+DELTAU
    DO209I=4,NCOEF
209 SGALP=SGALP+GALP(I)*GALP(I) (GRADIENT ALPHA SQUARED)
    WRITE(3,40)ALP
    DO300I=4,NCOEF
    CHG=ALP*GALP(I)/(SGALP*GAMMA) (CALC. DELTA A(I))
300 A(I)=A(I)-CHG                  (CHANGE A(I))
    IF(N.LT.2) GO TO 230
    IF(N.LT.(NTRY-5)) GO TO 150
230 WRITE(3,20)
    WRITE(3,17)(I,A(I),I=1,NCOEF)
150 CONTINUE                       (END ITERATION)
    WRITE(2,19)(A(I),I=1,NCOEF)
    STOP
    END
```

SOME EXACT SOLUTIONS TO APPROXIMATE
LINEAR AND NON-LINEAR VIBRATION PROBLEMS

by

THAD ALAN KING

B. S., Kansas State University, 1961

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1966

The report presents a procedure for determining exact solutions to approximate linear and non-linear vibration problems. Two methods are employed, Runge-Kutta and steepest descent.

Three example problems are given for a mass-damper-spring arrangement. The first, a linear system, is compared to the actual problem. The results are adequate for most engineering work. The second and third cases are non-linear. For these cases the results are not as accurate but demonstrate the abilities of the procedure.