

Integration Formulas and Schemes Based on g -Splines

By George D. Andria, George D. Byrne and David R. Hill

Abstract. Numerical integration formulas of interpolatory type are generated by the integration of g -splines. These formulas, which are best in the sense of Sard, are used to construct predictor-corrector and block implicit schemes. The schemes are then compared with Adams-Bashforth-Adams-Moulton and Rosser schemes for a particular set of prototype problems. Moreover, an improved error bound for linear multistep formulas based on g -splines and a comparison of L^2 norms of Peano kernels for Adams and natural spline formulas are given.

1. Introduction. This paper is a sequel to [5], where a systematic technique for constructing linear multistep formulas based on g -splines was given. Here, we shall give examples of g -spline based formulas, construct numerical integration schemes with them, and present some numerical results for the solution of problems of the type

$$(1.1) \quad \begin{aligned} y'(x) &= f(x, y(x)), & x \in I, \\ y(a) &= \eta, \end{aligned}$$

where $I = [a, b]$.

There are indeed numerous techniques or formulas for solving (1.1) that are based on piecewise polynomial (or spline) approximation ([2], [5], [6], [9], [10], [14], [15], [19], [20], [21], [23], [25]). Several of these references give no numerical results. Of those remaining, only [19] alludes to an algorithm [18] and its comparison with a standard discrete variable method. (See [17] for a summary.) Thus, we are attempting to alleviate the shortcoming of a lack of formulas and numerical results in at least one of the above references [5].

Our format is as follows. In Section 2, we begin with several Hermite-Birkhoff (H-B) interpolation problems, construct the resulting g -spline interpolants to f , and generate families of integration formulas [5]. Each formula is therefore best in the sense of Sard [23]. These formulas are then used to construct PE(CE)ⁿ schemes [12] and compared with Adams-Bashforth-Adams-Moulton (AA) schemes of the same type and same order in Section 3. The basis for this comparison is a certain set of prototype problems for which fixed order and uniform step size were used. The AA schemes were chosen because of their well-known worth [16, p. 230]. In Section 4, some block implicit schemes [3] are discussed and compared with Rosser's Runge-Kutta for all seasons or block Runge-Kutta (BRK) [22]. The basis for comparison is similar to that outlined above. The motivation for using BRK is that these formulas are based on Lagrangian interpolation and several orders of formulas are available.

Although our general approach is that of discrete variable techniques, the discrete variable solution can be used to construct a g -spline (or piecewise g -spline) approxi-

Received July 14, 1972.

AMS (MOS) subject classifications (1970). Primary 65L05.

Copyright © 1973, American Mathematical Society

mant to f . This continuous variable approximant can be integrated and used to change step size or it could be used as a global approximant to Y , the exact solution of (1.1) [5].

With the exception of Table 11.0, all of the tables referenced in this paper appear in the microfiche supplement of this issue of MOC.

2. Some Integration Formulas Based on g -Splines. The basic idea here is to construct linear multistep formulas of the form

$$(2.1) \quad y_{n+u} - y_{n+q} = \sum_{(i,j) \in e} B_{ij} h^{i+1} f^{(i)}(x_{n+i}, y_{n+i}),$$

where $e = \{(i, j)\}$ is the set of N interpolatory conditions for the m -poised H-B interpolation problem. Also, y_i denotes an approximant to $Y(x_i)$, the exact solution of (1.1) evaluated at $x_i = a + jh$. If

$$(2.2) \quad \phi(s) = f(x_n + sh, Y(x_n + sh)), \quad s \in [0, k],$$

then we seek $S_m \in C^r[0, k]$ such that

$$(2.3) \quad S_m^{(i)}(i) = \phi^{(i)}(i).$$

In particular, we shall assume that e has been so chosen that i takes on each of the values $0, 1, \dots, k$ one or more times and that $j = p$ for at least one element $(i, j) \in e$. By stating that the H-B problem is m -poised, we mean that if $P \in \Pi_{m-1}$ where Π_{m-1} is the class of polynomials of degree $m - 1$ or less, and if $P^{(i)}(i) = 0$ for $(i, j) \in e$, then $P(s) \equiv 0$. We shall also require that $p < m < N$.

The fundamental g -splines L_{ij} can be described by

$$(2.4) \quad L_{ij}^{(r)}(t) = \delta_{it} \delta_{jr}, \quad \text{if } (t, r) \in e,$$

so that the natural g -spline interpolant (g -spline interpolant) to ϕ can be written as

$$(2.5) \quad S_m(s) = \sum_{(i,j) \in e} L_{ij}(s) \phi^{(i)}(i)$$

for the set of knots $\Delta = \{0, 1, \dots, k\}$.

The definition of a g -spline can easily be given if we introduce the augmented incidence matrix $E^* = [\epsilon_{ij}^*]$ ($i = 0, 1, \dots, k; j = 0, 1, \dots, m - 1$) whose last $m - p - 1$ columns are filled with zeros and whose remaining elements are zero or one, according as to whether $(i, j) \notin e$ or $(i, j) \in e$.

Definition 2.1 [25, Definition 3, p. 213]. A function S_m is called a natural g -spline for the set of knots $\Delta = \{Z_1, Z_2, \dots, Z_k\}$, the matrix E^* , and order m , provided that it satisfies the following conditions:

- (i) $S_m \in \Pi_{2m-1}$ in (Z_i, Z_{i+1}) for $i = 0, 1, \dots, k - 1$;
- (ii) $S_m \in \Pi_{m-1}$ in $(-\infty, Z_0)$ and in $(Z_k, +\infty)$;
- (iii) $S_m \in C^{m-1}(-\infty, +\infty)$; and
- (iv) if $\epsilon_{ij}^* = 0$, then $S_m^{(2m-i-1)}$ is continuous at Z_i .

We now assume that $f \in W_2^m[I]$, the collection of all real-valued functions g defined on I with $g^{(m-1)}$ absolutely continuous and $g^{(m)} \in L^2[I]$. As a consequence, if f is replaced by S_m , (2.5), and

$$B_{ij} = \int_q^u L_{ij}(s) ds, \quad (i, j) \in e,$$

then

$$\begin{aligned}
 |Rf| &= \left| \int_{x_n+q}^{x_{n+u}} f(x, Y(x)) dx - \sum_{(i,j) \in \epsilon} h^{j+1} B_{ij} f^{(i)}(x_{n+i}, Y(x_{n+i})) \right| \\
 (2.6) \quad &= h^{m+1} \left| \int_{\iota} K(s) f^{(m)}(x_n + sh, Y(x_n + sh)) ds \right| \\
 &\leq h^{m+1} \|K\|_{L^2(\iota)} \|f^{(m)}\|_{L^2(\iota)},
 \end{aligned}$$

where

$$\begin{aligned}
 \iota &= \{s : 0 \leq s \leq \max(k, u)\}, \\
 \|f^{(m)}\|_{L^2(\iota)} &= \left[\int_{\iota} [d^m f(x_n + sh, Y(x_n + sh))/dx^m]^2 ds \right]^{1/2}, \\
 (2.7) \quad K(s) &= \left\{ \int_q^u [(Z - s)_+^{m-1} / (m - 1)!] dZ \right. \\
 &\quad \left. - \sum_{(i,j) \in \epsilon} [B_{ij}(i - s)_+^{m-i-1} / (m - j - 1)!] \right\},
 \end{aligned}$$

and $s \in \iota$ with

$$\begin{aligned}
 (Z - s)_+^i &= 0, & \text{if } Z \leq s, \\
 &= (Z - s)^i & \text{if } Z > s.
 \end{aligned}$$

Moreover, $\|K\|_{L^2(\iota)}$ is a minimum subject to the m constraints

$$\int_q^u \frac{x^u}{u!} dx = \sum_{(i,j) \in \epsilon} [B_{ij} / (u - j)!], \quad u = 0, 1, \dots, m - 1.$$

In short, (2.1) is a best approximation to $\int_{x_q}^{x_u} f(x, Y(x)) dx$ in the sense of Sard. It is to be noted that the error bounds are $O(h^{m+1})$ in (2.6) rather than $O(h^{m+1/2})$ as in [5, (2.8)]. This improvement results from using $\|f^{(m)}\|_{L^2(\iota)}$ instead of $\|f^{(m)}\|_{L^1(I)}$. See [2], [5], [25] for more details and consequences.

The various formulas we have generated are presented in the following manner. For $N = 1, 2, \dots, 8$, Table N.1 gives the fundamental g-splines L_{ij} for a given H-B problem. From these data, (2.3) and (2.5), the g-spline interpolant to ϕ (or f) can be constructed. For the same H-B problem, Table N.2 lists the coefficients B_{ij} for various intervals $[q, u]$ for the construction of formulas of type (2.1). The entries in the tables can be explained by the example $.1625(+01) = (.1625)(10^{+01})$.

The data in Tables 1.1–8.2 were generated on an IBM 360/65 using FORTRAN IV (G1) and the FORTRAN IV (Mod I) library. This particular library rounds data rather than chopping it. All calculations were done in double precision and then reduced to a twelve digit mantissa and all integrations were carried out exactly. We chose to use Schoenberg's algorithm [25] to construct the L_{ij} rather than to construct S_m directly.

We hasten to point out that some of these data have been given by Duris [9], who generated quadrature formulas by means of generalized inverses. This technique allowed $p = 0$ only but it did permit $m \leq N$. However, the case $m = N$ gives best formulas in the sense of Newton-Cotes and is not treated further here. Duris

treated several choices of N for each m , for m even only, and $[q, u] = [0, k], [k, k + 1], [k - 1, k]$.

Although our philosophy and methods for generating formulas differ from Duris', the formulas that should agree with his do so, except possibly in the last digit. But for Tables 6.1–8.2, we consider $m = N - 1, p = 0$. Our motive for considering these natural splines ([10], [24]) is the construction of PE(CE) $^\mu$ and block implicit schemes to compete with AA and BRK schemes.

For the sake of comparison, Tables 9.1–9.2 are the exactly computed analogs of Tables 3.1–3.2. The splines in Table 9.1 and the formulas for $[q, u] = [3, 4], [4, 5]$ in Table 9.2, also appear in [5].

3. PE(CE) $^\mu$ Schemes. In this section, several PE(CE) $^\mu$ schemes based on g -splines are compared with similar Adams-Bashforth-Adams-Moulton (AA) schemes. By a PE(CE) $^\mu$ scheme, we mean that a predictor or open formula is used to *predict* the value y_n , then f is *evaluated* at (x_n, y_n) . Next, a corrector or closed formula is used to *correct* y_n and f is *evaluated* again. The last two steps are executed exactly μ times [12].

In Tables 10.1–10.5, AA m denotes the m th order AA predictor-corrector pair. These formulas can be found in [7, pp. 136–137, 143–144] and [16, pp. 152, 180–181]. In the same set of tables, SS m denotes the m th order predictor-corrector pair corresponding to the natural spline S_m . This spline is given in Table $(m - 1).1$, while the formulas appear in Table $(m - 1).2$ with $[q, u] = [m, m + 1], [m - 1, m]$ for the predictor and corrector, respectively. Here, $m = 2, 3, \dots, 6$. We also tested some formulas for which $p \neq 0$. The symbol GG4 is used to denote the predictor-corrector pair of Table 7.2 corresponding to $[q, u] = [2, 3], [1, 2]$.

Our reasons for considering AA and AA-like schemes can be better explained as follows. The characteristic polynomial associated with (2.1) and (1.1), where $\partial f(x, y)/\partial y = \lambda$, λ a constant, is

$$(3.1) \quad \zeta^{n+u} - \zeta^{n+q} = \sum_{(i,j) \in e} B_{ij} h^{i+1} \lambda^j \zeta^{n+i}.$$

In particular, if $h = 0$, (3.1) reduces to

$$(3.2) \quad \zeta^r - \zeta^{r-1} = 0, \quad r \geq 2,$$

for all of the predictors and correctors considered here. Such formulas are *strongly stable*, since exactly one root of (3.2) has modulus one and the other roots are of modulus less than one ([4, p. 64], [11, p. 242]). If the B_{ij} were computed exactly, then (3.1) would be exact for $Y \in \Pi_m$. If $m \geq 2$, then (3.2) is *consistent*. It can be shown [1, Theorem 11, p. 107] that a strongly stable, consistent formula is absolutely stable. By *absolutely stable*, we mean there is a region \mathcal{R} in the complex $h\lambda$ plane, including $h\lambda = 0$, for which no root of (3.1) exceeds one in modulus and the roots of modulus one are simple. Absolute stability guarantees that $|Y(x_n) - y_n|$ does not grow without bound for $h\lambda \in \mathcal{R}$. These same notions carry over to PE(CE) $^\mu$ schemes [8].

The problems solved along with $\partial f/\partial y$ and their exact solutions were [13]

$$(3.3) \quad \begin{aligned} y'(x) &= -y(x), & y(0) &= 1, \\ \partial f(x, y)/\partial y &= -1, & Y(x) &= \exp(-x), \end{aligned}$$

$$\begin{aligned}
 (3.4) \quad & y'(x) = - [y(x)]^2/2, & y(0) = 1, \\
 & \partial f(x, y)/\partial y = -3y^2/2, & Y(x) = (x + 1)^{-1/2}, \\
 (3.5) \quad & y'(x) = y(x) \cos x, & y(0) = 1, \\
 & \partial f(x, y)/\partial y = \cos x, & Y(x) = \exp(\sin x), \\
 (3.6) \quad & y'(x) = [y(x)/4][1 - y(x)/20], & y(0) = 1, \\
 & \partial f(x, y)/\partial y = (1/4) - (y/40), & Y(x) = 20[1 + 19 \exp(-x/4)]^{-1}, \\
 (3.7) \quad & y'(x) = [y(x) - x]/[y(x) + x], & y(0) = 4, \\
 & \partial f(x, y)/\partial y = 2x/[y + x]^2, & R(\theta) = 4 \exp(\pi/2 - \theta),
 \end{aligned}$$

where $R^2 = x^2 + Y^2(x)$, $\theta = \tan^{-1}[Y(x)/x]$. Other problems were also solved with comparable numerical results. The data in Tables 10.1–10.5 were compiled on a PDP-10 with double precision and FORTRAN IV, version 22.8, at the Computer Center, University of Pittsburgh. Exact starting values were used in every instance and the true solutions Y were evaluated by the use of system routines.

These tables indicate that the m th order spline schemes perform better than the m th order Adams schemes, but not as well as the $(m + 1)$ th order Adams schemes. One notable exception was SS6 for which the data were poor, apparently because of the mantissa length of the B_{i_0} . The natural spline formulas require no more derivative evaluations than the Adams formulas, but SS m requires just as much data as AA $m + 1$. The extra data can be thought of as being used to reduce $\|K\|_{L^2(I_1)}$. For completeness, Table 10.6 gives a comparison of these constants in the local discretization (truncation) error bounds.

4. Block Implicit Schemes. Consider a set of k formulas of order m of the type

$$(4.1) \quad y_{n+u} - y_n = h \sum_{i=0}^k B_{i_0}(u)f(x_{n+i}, y_{n+i}), \quad u = 1, 2, \dots, k.$$

As an example, the formulas in Table 9.2 corresponding to $[q, u] = [0, 1], \dots, [0, 4]$ and $k = m = 4$ would constitute such a set. We can rewrite (4.1) as

$$(4.2) \quad \mathbf{y}_\nu = h\mathbf{B}\mathbf{F}(\mathbf{y}_\nu) + h\mathbf{b}f(x_n, y_n) + \mathbf{e}y_n,$$

where

$$\begin{aligned}
 \mathbf{B} &= [B_{i_0}(u)], \quad \text{for } i = 1, 2, \dots, k, u = 1, 2, \dots, k; \\
 \mathbf{y}_\nu &= (y_{n+1}, y_{n+2}, \dots, y_{n+k})^T; \\
 \mathbf{F}(\mathbf{y}_\nu) &= [f(x_{n+1}, y_{n+1}), f(x_{n+2}, y_{n+2}), \dots, f(x_{n+k}, y_{n+k})]^T; \\
 \mathbf{b} &= [B_{0_0}(u)], \quad u = 1, 2, \dots, k; \\
 \mathbf{e} &= (1, 1, \dots, 1)^T;
 \end{aligned}$$

and

$$n = k\nu.$$

For appropriate formulas, if y_n is given and some estimate of y_ν is made, (4.2) can

be solved to obtain a block of approximates $(y_{n+1}, \dots, y_{n+k})$. Elsewhere [3], we have dealt with systems somewhat more general than (4.2). There, the coefficient matrix of y , was not required to be the identity matrix and, hence, B and e were altered. In general, numerical results were not quite as good for those formulas with $S \neq I$ as for those with $S = I$.

There are other block implicit schemes and several techniques for applying them ([3], [22], [26]). Here, we are primarily interested in the comparison of the m th order Runge-Kutta for all seasons [22] and the m th order block methods based on natural splines. The following summarizes our mode of operation. Nonlinear Gauss-Seidel iteration was used to solve (4.2). When two consecutive iterates satisfied

$$\|y_v^{[q]} - y_v^{[q-1]}\| \leq T$$

for a prescribed constant T , $y_v^{[q]}$ was taken as the solution of (4.2). Otherwise, the Gauss-Seidel technique was applied again. In general, T has been so chosen that it is the largest number for which E cannot be reduced by further reduction of T , where $E = \max\{|Y(x_i) - y_i|\}$. In Tables 11.1–11.5, the initial guess $y_v^{[0]}$ was obtained by using a linear multistep predictor of order m . For the block natural spline scheme of order m , BSm , the formulas corresponding to (4.2), were taken from Table $(m - 1).2$ with $[q, u] = [0, 1], \dots, [0, m]$ and the predictor was taken from that table with $[q, u] = [m, m + 1]$. Also, $BRKm$ denotes the m th order Rosser scheme, which uses an m th order predictor, given in [22], to start. In Tables 11.1–11.5, R denotes the average number of applications of the nonlinear Gauss-Seidel method per block. This, together with Table 11.0 indicates the amount of work required to solve a problem.

Example 4.1. From Table 11.1, we find that for $h = 2^{-4}$, $BRK3$ requires 4 function evaluations per block and 160 blocks to solve the problem, for a total of about 640 evaluations of f . For $h = 2^{-4}$, $BS3$ requires about $3 + 3(2.1) = 9.3$ function evaluations per block and 107 blocks or about 995 evaluations of f to solve the same problem with less than half as much error.

Let R_R and R_S denote the average number of Gauss-Seidel iterates per block for the m th order Rosser and m th order natural spline schemes, respectively. If the total number of solution points, x_1, x_2, \dots, x_N , and the total number of function evaluations are to be the same for the two schemes, then

$$(4.3) \quad R_R = R_S.$$

However, this type of evaluation indicates nothing about the error incurred. One

TABLE 11.0. *A comparison of work required*

Scheme	Number of Function Evaluations per Block		
	Predictor	One Gauss-Seidel Iteration	Number of New Data Point Per Block
BSm	m	m	m
$BRKm$	$m - 1$	$m - 1$	$m - 1$

could perform a cost analysis for each problem, a particular machine, etc. But how much is accuracy worth? We suggest that the ratio

$$(4.4) \quad \omega = (1 + R_S)E_S h_R / (1 + R_R)E_R h_S,$$

where E_R and E_S denote the error of the Rosser and spline schemes, takes into account both work and accuracy and it appears to be realistic for certain applications. For $\omega < 1$, the block implicit natural spline method of order m_S can be regarded as better than the Rosser method of order m_R . Of course, h_R and h_S denote step size.

Example 4.2. Using the data of Example 4.1, we find that

$$\omega = \frac{(1 + 2.1)(.16 \cdot 10^{-6})2^{-4}}{(1 + 1.0)(.38 \cdot 10^{-6})2^{-4}} \doteq .65.$$

If h_R is changed to 2^{-5} , then

$$\omega = \frac{(1 + 2.1)(.16 \cdot 10^{-6})2^{-5}}{(1 + 1.0)(.30 \cdot 10^{-7})2^{-4}} \doteq 4.1.$$

Of course, the Rosser formula has given twice as much data in the second case! If $h_R = 2^{-4}$ and $h_S = 2^{-5}$, then

$$\omega \doteq .09.$$

If ω is used as a comparison in Tables 11.1–11.5, the natural spline formulas compare very favorably with Runge-Kutta for all seasons. Other criteria could be used for comparison, and various values of T and R could be used.

For stability analyses, strategies for implementation, and convergence proof for block implicit methods, see [3], [22], and [26].

The data in Tables 11.1–11.5 were obtained with the hardware and software described immediately after (3.7). Again, the loss of accuracy in the higher order spline data can be attributed to the use of twelve digit mantissas in the $B_{i,0}$.

5. Concluding Remarks. In Tables 1.1–9.2, several g -splines are given along with a variety of integration formulas which were obtained by the integration of these splines. Thus, the examples of [2], [3], [5], [23], [24], and [25] have been supplemented by the use of a g -spline generating algorithm which seems to be more general than that of [9]. We also note that the error bound for the integration formulas (2.6) is more satisfying than that of [5, (2.8)].

The integration formulas were next used to generate PE(CE) ^{μ} schemes and block implicit schemes. Tables are also given in which these integration schemes are compared with some others. As a consequence of these data, we feel that piecewise polynomial functions can indeed be used to construct efficient techniques for solving (1.1). Thus, some of the shortcomings of [15] and [17], such as excessive functional evaluations, use of derivatives of f , or stability problems, have been avoided. If S , a g -spline approximant to Y , is desired, the g -spline interpolants can be pieced together and integrated. As a consequence, $S \in C^1(-\infty, \infty)$ in general ([5], [14], [15]).

1. L. V. AHLFORS, *Complex Analysis, An Introduction to the Theory of Analytic Functions of One Variable*, McGraw-Hill, New York, 1953. MR 14, 857.
2. J. H. AHLBERG & E. N. NILSON, "The approximation of linear functionals," *SIAM J. Numer. Anal.*, v. 3, 1966, pp. 173-182. MR 36 #589.
3. G. D. ANDRIA, G. D. BYRNE & D. R. HILL, "Natural spline block implicit methods," *Nordisk. Tidskr. Informationsbehandling (BIT)*, v. 13, 1973. (To appear.)
4. I. BABUŠKA, M. PRÁGER & E. VITÁSEK, *Numerical Solution of Differential Equations*, Státní Nakladatelství Technické Literatury, Prague, 1964; English transl., Interscience, New York, 1966. MR 36 #6149; #6150.
5. G. D. BYRNE & D. N. H. CHI, "Linear multistep formulas based on g -splines," *SIAM J. Numer. Anal.*, v. 9, 1972, pp. 316-324.
6. E. D. CALLENDER, "Single step methods and low order splines for solution of ordinary differential equations," *SIAM J. Numer. Anal.*, v. 8, 1971, pp. 61-66.
7. F. CESCHINO & J. KUNTZMAN, *Numerical Solution of Initial Value Problems*, Dunod, Paris, 1963; English transl., Prentice-Hall, Englewood Cliffs, N.J., 1966. MR 28 #2639; MR 33 #3465.
8. P. E. CHASE, "Stability properties of predictor-corrector methods for ordinary differential equations," *J. Assoc. Comput. Mach.*, v. 9, 1962, pp. 457-468. MR 29 #738.
9. C. S. DURIS, *Optimal Quadrature Formulas Using Generalized Inverses*. Part II: *Sard "Best" Formulas*, Math. Rep. 70-71, Dept. of Mathematics, Drexel University, Philadelphia, 1970.
10. T. N. E. GREVILLE, "Spline functions, interpolation, and numerical quadrature," *Mathematical Methods for Digital Computers*. Vol. II, A. Ralston and H. S. Wilf (Editors), Wiley, New York, 1968, pp. 156-168.
11. P. HENRICI, *Discrete Variable Methods in Ordinary Differential Equations*, Wiley, New York, 1962. MR 24 #B1772.
12. T. E. HULL & A. L. CREAMER, "Efficiency of predictor-corrector procedures," *J. Assoc. Comput. Mach.*, v. 10, 1963, pp. 291-301. MR 27 #4367.
13. T. E. HULL, W. H. ENRIGHT, B. M. FELLER & A. E. SEDGWICK, *Comparing Numerical Methods for Ordinary Differential Equations*, Technical Report #29, Dept. of Computer Science, University of Toronto, 1971.
14. B. L. HULME, *One-Step Piecewise Polynomial Galerkin Methods for Initial Value Problems*, Technical Memo. SC-TM-710127, Sandia Laboratories, Albuquerque, N.M., 1971; *Math. Comp.*, v. 26, 1972, pp. 415-426.
15. B. L. HULME, "Piecewise polynomial Taylor methods for initial value problems," *Numer. Math.*, v. 17, 1971, pp. 367-381.
16. L. LAPIDUS & J. H. SEINFELD, *Numerical Solution of Ordinary Differential Equations*, Math. in Sci. and Engineering, vol. 74, Academic Press, New York, 1971. MR 43 #7073.
17. F. R. LOSCALZO, *An Introduction to the Application of Spline Functions to Initial Value Problems* (Proc. Sem., Math. Research Center, Univ. of Wisconsin, Madison, Wis., 1968), Academic Press, New York, 1969, pp. 37-64. MR 39 #2334.
18. F. R. LOSCALZO, *Numerical Solution of Ordinary Differential Equations by Spline Functions (SPLINDIFF)*, Technical Summary Report 842, Mathematics Research Center, U. S. Army, University of Wisconsin, Madison, Wis., 1968.
19. F. R. LOSCALZO & I. J. SCHOENBERG, *On the Use of Spline Functions for Solutions of Ordinary Differential Equations*, Technical Summary Report 723, Mathematics Research Center, U.S. Army, University of Wisconsin, Madison, Wis., 1967.
20. F. R. LOSCALZO & T. D. TALBOT, "Spline function approximation for solutions of ordinary differential equations," *Bull. Amer. Math. Soc.*, v. 73, 1967, pp. 438-442. MR 35 #1218.
21. F. R. LOSCALZO & T. D. TALBOT, "Spline function approximations for solutions of ordinary differential equations," *SIAM J. Numer. Anal.*, v. 4, 1967, pp. 433-445. MR 36 #4808.
22. J. B. ROSSER, "A Runge-Kutta for all seasons," *SIAM Rev.*, v. 9, 1967, pp. 417-452. MR 36 #2325.
23. A. SARD, *Linear Approximation*, Math. Surveys, no. 9, Amer. Math. Soc., Providence, R.I., 1963. MR 28 #1429.
24. I. J. SCHOENBERG, "On best approximation of linear operators," *Nederl. Akad. Wetensch. Proc. Ser. A*, v. 67 = *Indag. Math.*, v. 26, 1964, pp. 155-163. MR 28 #4284.
25. I. J. SCHOENBERG, "On the Ahlberg-Nilson extension of spline interpolation: the g -splines and their optimal properties," *J. Math. Anal. Appl.*, v. 21, 1968, pp. 207-231. MR 36 #6849.
26. L. F. SHAMPINE & H. A. WATTS, "Block implicit one-step methods," *Math. Comp.*, v. 23, 1969, pp. 731-740. MR 41 #9445.