Numerical Methods for Time-Dependent, Nonlinear Boundary Value Problems

W. E. CULHAM
JUNIOR MEMBER AIME
RICHARD S. VARGA

GULF RESEARCH & DEVELOPMENT CO.
PITTSBURGH, PA.
KENT STATE U.
KENT, OHIO

ABSTRACT

This paper presents and examines in detail extensions to the Galerkin method of solution that make it numerically superior to conventional methods used to solve a certain class of time-dependent, nonlinear boundary value problems. This class of problems includes the equation that describes the flow of a fully compressible fluid in a porous medium.

The Galerkin method with several different piecewise polynomial subspaces and a non-Galerkin method specifically employing cubic spline functions are used to approximate the solution of a nonlinear parabolic equation with one spatial variable. With a known analytic solution of the problem, the accuracies of these approximations are determined and compared with conventional finite-difference approximations. Specifically, the various methods are compared on the basis of the amount of computer time necessary to achieve a given accuracy, as well as with respect to the order of convergence and computer core storage required. These tests indicate that the higher-order Galerkin methods require the least amount of computer time for a given range of accuracy.

INTRODUCTION

The purpose of this paper is to outline in detail the application of the Galerkin method, employing piecewise polynomials, to solve nonlinear-boundary-value problems and compare the computational efficiency of the Galerkin method with more conventional numerical methods. Numerical methods compared with the Galerkin technique include a non-Galerkin method that utilizes cubic spline interpolation and the conventional finite-difference methods. Four conventional time approximations were also studied in conjunction with the above mentioned space discretization methods.

In an earlier paper, Price and Varga[2] showed theoretically that higher-order approximations to certain semilinear convection-diffusion equations were possible by means of Galerkin techniques, but complete numerical results for such approximations were not given. Also, in a paper that introduced the Galerkin method to the petroleum industry, Price *et al.*[1] demonstrated that higher-order approximations were far superior numerically to the conventional methods used to solve certain linear convection-diffusion type problems. Jennings,[14] Douglas and Dupont[13] and Douglas *et al.*[15] have considered the application of Galerkin methods to various nonlinear problems, but again complete numerical results, including comprehensive comparisons with existing numerical methods, were not given. Thus, in addition to presenting some new and computationally efficient Galerkin formulations for nonlinear problems and numerically demonstrating their higher-order accuracies, it was also desirable to test these methods to determine if they also exhibited the same superiority in regard to computational efficiency as was demonstrated for the Galerkin methods applied to linear problems. If so, then the Galerkin technique could prove to be an important advancement toward developing faster numerical models for field application.

To test and compare each method of solution, a problem involving the nonlinear gas-flow equation in one spatial variable with a specific volumetric source term was chosen, for which a closed-form or analytic solution was known. Using this particular problem and its analytic solution, it was possible to determine numerically the order of convergence of each method, to compare each method on the basis of computer time expended to obtain a given accuracy, and to compare each method with respect to computer core storage required. In addition, the experimental data were used to define "consistent quadrature" and "consistent interpolation" schemes for the Galerkin methods. Finally, it was possible to formulate conclusions regarding the computational efficiency of the four time approximations investigated.

Original manuscript received in Society of Petroleum Engineers office Jan. 2, 1970. Revised manuscript received April 1, 1971. Paper (SPE 2806) was presented at SPE Second Symposium on Numerical Simulation of Reservoir Performance, held in Dallas, Feb. 5-6, 1970.

[1]References given at end of paper.

PROCEDURE OF INVESTIGATION

BOUNDARY VALUE PROBLEM

The problem chosen to serve as a basis for comparing the various numerical methods listed below describes the transient flow of a real gas through a porous medium. The differential equation considered for this study is nonlinear and includes a volumetric source term. The particular volumetric source term used in this study was so constructed that it provides local areas of the system with a significantly stronger source potential than the surrounding areas. The gas flow equation considered is:

$$\frac{\partial}{\partial x}\left(\alpha(p)\frac{\partial p}{\partial x}\right)+\frac{\bar{B}T}{M}S(x,t)=\beta(p)\frac{\partial p}{\partial t};$$

$$0<x<L,\ t>0 \quad \ldots\ldots\ldots\ldots (1)$$

where

$$\alpha(p)=\frac{kp}{z\mu} \quad \ldots\ldots\ldots\ldots (2)$$

and

$$\beta(p)=\frac{\phi A}{z}\left(1-\frac{p}{z}\frac{dz}{dp}\right). \quad \ldots\ldots (3)$$

Initial and boundary conditions used in this paper are:

$$p(x,0)=p_o;\ 0\le x\le L \quad \ldots\ldots (4)$$

and

$$\left.\frac{\partial p}{\partial x}\right|_{x=0}=0;\ t>0 \quad \ldots\ldots\ldots (5)$$

$$p(L,t)=p_o\left(1-\alpha_1\gamma t e^{-\gamma t}\right);\ t>0 \quad \ldots (6)$$

The gas deviation factor and viscosity in the above equations are treated as pressure dependent functions and the absolute permeability is treated as a spatially dependent variable. The actual values employed for the gas deviation factor and viscosity were typical of a 0.7 gravity gas at a temperature of 233°F. The initial pressure was set at 4,150 psi and practical field values were employed for the other physical properties of the hypothetical reservoir. The solution to this problem is

$$p(x,t)=p_o\left[\alpha_2\left(1-e^{-\alpha_3\gamma t}\right)f(x)+\left(1-\alpha_1\gamma t e^{-\gamma t}\right)\right], \quad \ldots\ldots\ldots\ldots (7)$$

where

$$f(x)=\left(\frac{x}{L}\right)^2\left(0.25-\frac{x}{L}\right)\left(0.5-\frac{x}{L}\right)\left(0.75-\frac{x}{L}\right)\left(1.0-\frac{x}{L}\right), \quad \ldots (8)$$

and the source term employed in Eq. 1 is given by

$$S(x,t)=\frac{p_o M}{\bar{B}T}\left[\gamma\beta\left(\alpha_2\alpha_3 f(x)e^{-\alpha_3\gamma t}+\alpha_1 e^{-\gamma t}\right.\right.$$
$$\left.(\gamma t-1)-\alpha_2\left(1-e^{-\alpha_3\gamma t}\right)\right)\frac{df}{dx}\frac{\partial\alpha}{\partial x}$$
$$\left.-\alpha_2\alpha\left(1-e^{-\alpha_3\gamma t}\right)\frac{d^2f}{dx^2}\right]. \quad \ldots\ldots (9)$$

Pressure-dependent terms in Eq. 9 are given explicitly by the solution, Eq. 7.

Eqs. 1 through 6 with the source term defined by Eq. 9 were solved numerically using the methods defined in the next section.

NUMERICAL METHODS

Four Galerkin-type methods (to be identified by the basis functions used), a non-Galerkin method employing piecewise cubic spline interpolation, and the standard central finite-difference approximation were utilized to discretize the space variable of the subject problem. Four different time approximations were used with each of the space discretization methods. They are (1) backward time approximation,[3] (2) Crank-Nicolson time approximation,[3] (3) Lees' three-level time approximation,[11] and (4) modified backward time approximation.

GALERKIN-TYPE METHODS

A simple, straightforward application of the Galerkin method, employing continuous piecewise polynomial functions as basis functions to the boundary value problem of concern, results in a semidiscrete approximation,[3] which in matrix ordinary-differential-equation notation is:

$$B\frac{d\underline{C}}{dt}+A\underline{C}=\underline{S} \quad \ldots\ldots\ldots (10)$$

Additional details on this formulation can be obtained from papers by Price and Varga,[2] and Cavendish *et al.*[1,4] The unknown vector $\underline{C}$, in Eq. 10, represents the time-dependent coefficients in the following approximation to the solution of Eqs. 1 through 6:

$$\hat{p}(x,t)=\sum_{i=1}^{m}c_i(t)w_i(x), \quad \ldots\ldots\ldots (11)$$

where m is the dimension of a particular subspace spanned by the m basis functions $w_i(x)$, $i=1,2,\ldots m$.

The vector $\underline{S}$ in Eq. 10 involves the source term and has elements defined by

$$S_i = \frac{\bar{B}T}{M} \int_o^L S(x,t)\, w_i(x)\, dx; \quad 1 \le i \le m \qquad \ldots\ldots\ldots\ldots (12)$$

and the mxm Matrices B and A have elements defined by

$$b_{ij} = \int_o^L \beta(\hat{p}(x,t))\, w_i\, w_j\, dx; 1 \le i,j \le m, \qquad \ldots\ldots\ldots\ldots (13)$$

and

$$a_{ij} = -\left[w_i\, \alpha(\hat{p}(x,t)) \frac{dw_j}{dx} \right]\Bigg|_{x=o}^{x=L} + \int_o^L \alpha(\hat{p}(x,t)) w_i' w_j'\, dx; \quad 1 \le i,j \le m. \qquad \ldots\ldots\ldots\ldots (14)$$

A modification, the importance of which will be demonstrated numerically later in the paper, to the elements of the source vector $\underline{S}$ and coefficient matrices of Eq. 10 will now be outlined. This modification involves interpolation of the source term and the pressure-dependent coefficients of the differential equation. That is, define $\hat{\alpha}$, $\hat{\beta}$ and $\hat{S}$ as the interpolates of $\alpha(\hat{p})$, $\beta(\hat{p})$ and $S(x,t)$ such that

$$\hat{\alpha} = \sum_{k=1}^{r} d_k(t)\, v_k(x)$$

$$\hat{\beta} = \sum_{k=1}^{r} f_k(t)\, v_k(x)$$

and

$$\hat{S} = \sum_{k=1}^{r} g_k(t)\, v_k(x).$$

Employing these three equations, the defining equations for the elements of the coefficient matrices and the source vector become:

$$b_{ij} = \sum_{k=1}^{r} f_k(t) \int_o^L v_k(x)\, w_i(x)\, w_j(x)\, dx; \quad 1 \le i,j \le m \qquad \ldots\ldots\ldots (15)$$

$$a_{ij} = -\left\{ w_i\, \alpha(\hat{p}) \frac{dw_j}{dx} \right\}\Bigg|_{x=o}^{x=L} + \sum_{k=1}^{r} d_k(t) \cdot \int_o^L v_k(x)\, w_i'(x)\, w_j'(x)\, dx; \quad 1 \le i,j \le m \qquad \ldots\ldots\ldots\ldots\ldots\ldots (16)$$

and

$$S_i = \frac{\bar{B}T}{M} \sum_{k=1}^{r} g_k(t) \int_o^L v_k(x)\, w_i(x)\, dx; \quad 1 \le i \le m. \qquad \ldots\ldots\ldots\ldots (17)$$

In contrast to the integrands in Eqs. 12 through 14, the integrands of the three integral terms above are strictly functions of the variable of integration and thus can be integrated exactly. Generation of the time-dependent parameters d_k, f_k and g_k, $1 \le k \le r$ depends on the method of interpolation employed. Four methods, cubic and quintic spline interpolation, cubic Hermite interpolation, and an integral least-squares procedure, all employing continuous piecewise basis elements, v_k, were studied. This approach, where the elements of the source vector and coefficient matrices are defined by Eqs. 15 through 17, will subsequently be referred to as the modified Galerkin method.

The choice of a particular set of basis functions to represent the solution in the form of Eq. 11 determines the dimension m of the previously mentioned matrices and also fixes the band widths of these matrices. Four different sets of basis functions were utilized in this study.

In the following description let $\Delta: 0 = x_1 < x_2 < \ldots < x_{N+1} = L$ denote any partition of $[0,L]$ with grid points x_i. Since the defining equations for the basis functions summarized below are readily available elsewhere,[5,6,7] they will not be presented here.

Smooth Linear Hermite Space $H^{(1)}(\Delta)$

For a fixed partition Δ of $[0, L]$ the functions of $H^{(1)}(\Delta)$ are continuous functions defined on $[0, L]$, which are linear on each subinterval (x_i, x_{i+1}) of $[0, L]$. It is possible to select a basis, $t_i(x)$, $i = 1, 2 \ldots, N+1$, called Chapeau functions,[7] such that the support of t_i is contained in the interval (x_{i-1}, x_{i+1}). Because of this, there is just one unknown per mesh point, and the resulting Matrices A and B are tridiagonal.

Smooth Cubic Hermite Space $H^{(2)}(\Delta)$

The elements of the smooth cubic Hermite space $H^{(2)}(\Delta)$ are functions $w(x)$, which are continuously differentiable on $[0, L]$, such that $w(x)$ is a cubic polynomial on each subinterval (x_i, x_{i+1}) of $[0, L]$, Because we can in essence assign the values of

$w(x_i)$ and $w'(x_i)$ at each x_i, such functions form a $2(N+1)$-dimensional subspace of $C^1[0, L]$. Moreover, it is possible to select a basis for $H^{(2)}(\Delta)$, such that the corresponding Matrices A and B have band widths of seven.[7]

Nonsmooth Cubic Hermite Space $H(\Delta, 1, 4)$

The elements of the nonsmooth cubic Hermite space are functions $w(x)$, which are only continuous on $[0, L]$, such that $w(x)$ is a cubic polynomial on each subinterval (x_i, x_{i+1}) of $[0, L]$. This space has dimension $3N+1$. Again, it is possible to select a basis for $H(\Delta, 1, 4)$[7] such that the corresponding Matrices A and B have band widths of seven.

Cubic Spline Space $Sp^{(2)}(\Delta)$

The elements of the cubic spline space $Sp^{(2)}(\Delta)$ are functions $w(x)$, which are *twice* continuously differentiable on $[0, L]$, such that $w(x)$ is a cubic polynomial on each subinterval (x_i, x_{i+1}) of $[0, L]$. This space has dimension $N+3$, and by choosing a particular basis for $Sp^{(2)}(\Delta)$, the Matrices A and B have again a band width of seven.[7]

Although no mention of boundary conditions was included in the above discussion, it is necessary to make modifications to Matrices A and B to take these conditions into account. Flux-type boundary conditions (e.g., Eq. 5) can be accounted for in a natural manner by utilizing the first term in the definition of a_{ij} given by Eq. 14. Herbold[5] outlines a procedure for treating specified potential-type boundary conditions when cubic spline basis functions are employed. Although the example given by Herbold involves homogeneous boundary conditions, his approach can be readily extended to nonhomogeneous conditions. Modifications needed for basis functions other than cubic spline functions are straightforward.

The initial set of coefficients, $c_i(0)$, $i = 1, 2, \ldots, m$ is generated by requiring that

$$\left\langle \hat{p}(x,0), w_j(x) \right\rangle = \left\langle p(x,0), w_j(x) \right\rangle \qquad j = 1, 2, \ldots m, \quad . \quad . \quad . \quad (18)$$

where

$$\hat{p}(x,0) = \sum_{i=1}^{m} c_i(0)\, w_i(x) \quad . \quad . \quad . \quad . \quad (19)$$

and $\langle , \rangle$ denotes the usual L_2-inner product on $[0, L]$. Thus Eq. 18 represents an integral least-squares approximation of the initial data by the piecewise set of functions $w_i(x)$, $1 \leq i \leq m$.

NON-GALERKIN CUBIC SPLINE INTERPOLATION

This approach was first suggested by Albasiny and Hoskins[8] for solving a linear two-point boundary value problem involving a second-order ordinary differential equation. The method has been modified to handle Eqs. 1 through 6*. Details of this modification are presented in the Appendix.

This method[9] utilizes the following cubic spline polynomial to interpolate to the pressure at time level $t_{n+1} = (n+1)\Delta t$ at the uniformly spaced grid points x_i, $1 \leq i \leq N+1$:

$$\bar{S}(x) = M_{i-1}\frac{(x_i - x)^3}{6h} + M_i\frac{(x - x_{i-1})^3}{6h} + \left(\hat{p}_{i-1} - \frac{h^2}{6}M_{i-1}\right)\frac{(x_i - x)}{h} + \left(\hat{p}_i - \frac{h^2}{6}M_i\right)\frac{(x - x_{i-1})}{h}; \quad x_{i-1} \leq x \leq x_i, \quad . \quad . \quad . \quad . \quad (20)$$

where $M_i = \bar{S}''(x_i)$ and $\hat{p}_i = \hat{p}(x_i, t_{n+1})$. The requirement that the spline approximation satisfy the differential equation (Eq. A-1) at the grid points x_i, $1 \leq i \leq N+1$ furnishes, on using Eq. 20 and the continuity of first derivatives at the grid points, a set of relations that can be used to eliminate the unknowns M_i, $1 \leq i \leq N+1$. The final result, after invoking the boundary conditions, is a tridiagonal set of equations for the determination of $\hat{p}(x_i, t_{n+1})$, $1 \leq i \leq N+1$. These discrete values of pressure in turn can be used to explicitly determine M_i and thus generate a twice continuously differentiable solution in the form of Eq. 20 for the entire interval $[0, L]$.

A close examination of this method indicates that it is simply a procedure that simultaneously generates the solution $\hat{p}(x_i, t_{n+1})$, $1 \leq i \leq N+1$ and the cubic-spline-interpolation polynomial of these discrete values. The same results could be obtained by interpolating independently the discrete values of pressure generated by a finite-difference equation similar to the Numerov formula.[10]

CENTRAL-DIFFERENCE APPROXIMATIONS (CDA)

The CDA approximation is just the standard, second-order correct, central finite-difference approximation. It is readily available in the literature.[3]

DISCRETE APPROXIMATIONS IN TIME

As previously outlined, four common time approximations were utilized in this study. For the sake of convenience and to facilitate later discussions, these methods will be outlined by using Eq. 10.

Backward-Time Approximation (BKWD)

Employing the first-order correct backward-time approximation to fully discretize Eq. 10 and letting the subscript "n" denote the discrete time level $t_n = n\Delta t$ results in

*For this method the boundary condition at $x = 0$ was changed to one involving specification of pressure rather than a flux-type boundary condition.

$$\left[B_{n+1}+\Delta t A_{n+1}\right]\underline{C}_{n+1}=B_{n+1}\underline{C}_n+\Delta t\underline{S}_{n+1}, \quad \ldots\ldots (21)$$

where a subscript on a matrix implies that the elements that constitute the matrix are to be evaluated at that time level.

Crank-Nicolson Time Approximation (CN)

Employing the second-order correct Crank-Nicolson time approximation to fully discretize Eq. 10 results in:

$$\left[B_{n+\frac{1}{2}}+\frac{\Delta t}{2}A_{n+\frac{1}{2}}\right]\underline{C}_{n+1}=\left[B_{n+\frac{1}{2}}-\frac{\Delta t}{2}A_{n+\frac{1}{2}}\right]\underline{C}_n+\Delta t\underline{S}_{n+\frac{1}{2}} \ldots (22)$$

Lees' Three-Level Time Approximation (L3L)

Use of Lees' three-level, second-order correct time approximation to fully discretizing Eq. 10 results in:

$$\left[B_n+\frac{2\Delta t}{3}A_n\right]\underline{C}_{n+1}=\left[B_n-\frac{2\Delta t}{3}A_n\right]\underline{C}_{n-1}-\left[\frac{2\Delta t}{3}A_n\right]\underline{C}_n+2\Delta t\underline{S}_n. \quad \ldots\ldots (23)$$

It should be noted that this method centers the coefficients at time level n, and thus Eq. 23 is linear and can be solved directly for the set of coefficients at the $(n+1)$-st time level.

Modified Backward-Time Approximation (MBKWD)

This scheme uses the same matrix equation structure as the fully backward-time approximation, but employs dependent variables at the nth time level to evaluate the coefficient matrices. Thus Eq. 21 becomes:

$$\left[B_n+\Delta t A_n\right]\underline{C}_{n+1}=B_n\underline{C}_n+\Delta t\underline{S}_{n+1} \ldots (24)$$

Two of the above time approximations result in sets of nonlinear equations that must be linearized in some manner in order to obtain a solution. The following iteration scheme, demonstrated by applying it to Eq. 21, was employed

$$\left[B_{n+1}^{(k)}+\Delta t A_{n+1}^{(k)}\right]\underline{C}_{n+1}^{(k+1)}=B_{n+1}^{(k)}\underline{C}_n+\Delta t\underline{S}_{n+1}, \quad \ldots\ldots (25)$$

where the superscript "k" is an iteration number.

The elements of A and B were evaluated by using

$$\hat{p}^{(k)}(x,(n+1)\Delta t))=\sum_{i=1}^{m}c_i^{(k)}((n+1)\Delta t)w_i(x), \quad \ldots\ldots (26)$$

and the scheme is started by using

$$\hat{p}^{(0)}(x,(n+1)\Delta t)=\hat{p}(x,n\Delta t).$$

NUMERICAL EXPERIMENTATION

Numerical experimentation was designed to accomplish three basic tasks: (1) to define the condition needed to insure "consistent quadrature and interpolation schemes" in the Galerkin methods, (2) to determine the numerical order of convergence of all methods tested, and (3) to compare all methods from the standpoint of computer time expended to obtain a given accuracy.

Use of the term "consistent quadrature scheme" is connected with the approximation of the numerous integrals needed to generate the coefficient matrices in the basic Galerkin equations (i.e., Eqs. 10 and 12 through 14). As originally defined by Herbold,[5] a consistent quadrature scheme is a quadrature method that when applied to the various integrals in the Galerkin method, gives the same theoretical rate of convergence (see Eq. 30) as if the integrals had been calculated with infinite precision. In essence, this means the exponent a of Eq. 30 is left unchanged by a consistent quadrature routine. The constant $K_1(t)$ in Eq. 30 can be affected by the quadrature scheme employed in the Galerkin method. *For the purposes of this paper, "consistent quadrature schemes" are defined* as those schemes which require the least amount of numerical computation while preserving the order of convergence of the method of solution and giving, to machine accuracy, the least value of $K_1(t)$ in Eq. 30. Thus, this definition differs from Herbold's in that it is concerned not only with the rate of convergence, but also the magnitude of the coefficient multiplying $(\Delta x)^a$ in Eq. 30. In an analogous manner, a "consistent interpolation scheme", which is relevant when the modified Galerkin formulation as outlined by Eqs. 15 through 17 is used, *can be defined* as an interpolation procedure that requires the least amount of numerical computation while preserving the order of convergence of the method of solution and giving, to machine accuracy, the least value of $K_1(t)$ in Eq. 30.

To accomplish these tasks, an analytic solution to Eqs. 1 through 6 and a measure of accuracy were required. The analytic solution to the subject problem is presented as Eq. 7. The measure of accuracy chosen for this study was the $L\infty$ norm. Although other norms could certainly be used for measuring the accuracy of the numerical solution, the $L\infty$, or maximum norm, was chosen because it has a meaning that can be readily interpreted.

The $L\infty$ norm is accurately estimated by putting a uniform mesh on $[0, L]$ of mesh size $h=L/N_o$, where N_o is a large positive integer, and computing

$$E(t) \equiv \max_{0 \le j \le N_o} \left| \hat{p}(jh,t) - p(jh,t) \right|, \quad (27)$$

where $p(jh,t)$ is the analytic solution given by Eq. 7 and $\hat{p}(jh,t)$ is its numerical approximation. In the data presented in this paper (except for that of Table 3) N_o was chosen to be 500. A few test cases employing N_o=2,000 generated values of $E(t)$ only slightly different from the value obtained using N_o=500. Eq. 27 can be applied to continuous-solution methods such as the Galerkin method and the non-Galerkin cubic spline interpolation method, but the following version of Eq. 27 was employed for the CDA approximations:

$$E(t) \equiv \max_{1 \le i \le N+1} \left| \hat{p}(x_i, t) - p(x_i, t) \right| \quad (28)$$

With respect to the time approximations that generated nonlinear algebraic equations, thereby necessitating the implementation of the previously outlined iteration scheme, it was necessary to carry the iteration to the point where the error defined by Eqs. 27 or 28 did not materially change. It was found that, for the conditions under which the series of tests were conducted, three iterations (or inner loops) were sufficient per time step.

For each of the methods tested, all computations were carried out in double-precision arithmetic on the IBM 360 Model 85 Computer.

RESULTS

To determine the conditions that insure a consistent quadrature scheme, two sets of experiments were carried out. These experiments consisted of solving the boundary value problem of concern using the Galerkin approach employing Chapeau and smooth cubic basis functions and a time step sufficiently small to insure that the error, given by Eq. 27, could principally be attributed to the spatial error. With the time step and mesh size fixed, the boundary value problem was solved using Gaussian quadrature formulas of various orders of accuracy (i.e., 2, 4, 8, 12 point formulas) to set up the coefficients matrices of Eq. 10 with terms defined by Eqs. 12 through 14. Figs. 1 and 2 illustrate the effect of changing the order or accuracy of the quadrature scheme on the error $E(t)$.

The tests employed to examine the consistent interpolation concept involved solving the test problem using the Galerkin-cubic spline and the Galerkin-smooth cubic methods with an appropriately small time step. With the time step and mesh size fixed, the problem was solved by using interpolation schemes, employing interpolation partitions ranging from eight times finer to the same size as the solution mesh, to generate the coefficient matrices and source term of Eq. 10 as defined by Eqs. 15, 16 and 17, respectively. Figs. 3 and 4 illustrate the effect of interpolation mesh refinement on the error $E(t)$ for the modified Galerkin-cubic spline method of solution in conjunction with cubic spline interpolation. Fig. 5 illustrates the effect of interpolation mesh refinement on the error $E(t)$ for the modified Galerkin-smooth Cubic method of solution in conjunction with cubic Hermite interpolation. The term "mesh refinement factor" used in Figs. 3 and 5 is defined as follows. Let $\Delta_s: 0 = x_1 < x_2 < \ldots < x_{N+1} = L$ denote the basic partition of $[0, L]$ used in the solution of the problem. This mesh will subsequently be referred to as the solution mesh. The interpolation mesh is defined as the partition formed by subdividing each subinterval (x_i, x_{i+1}) of the solution mesh into RF equal intervals. RF is then defined as the mesh refinement factor and in this paper is a positive integer greater than or equal to one.

The spatial order of convergence of the methods tested was determined by plotting log $E(t)$ vs log Δx. The reason for employing a plot of this type is based on the following equation,[1] which is used to define the error given by Eqs. 27 or 28,

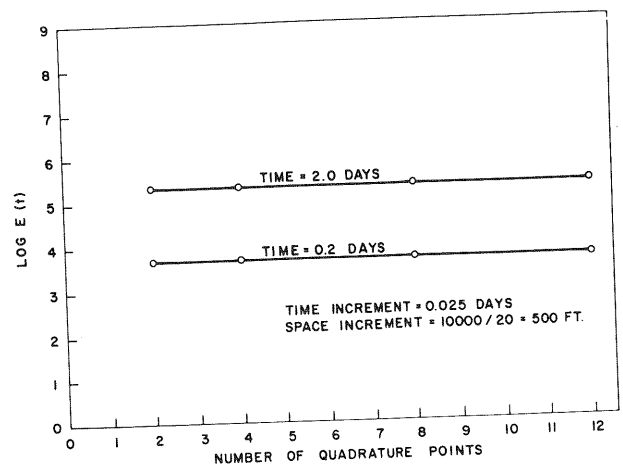

FIG. 1 — EFFECT OF ORDER OF QUADRATURE ON ERROR — BASIC GALERKIN METHOD USING CHAPEAU-BASIS FUNCTIONS.

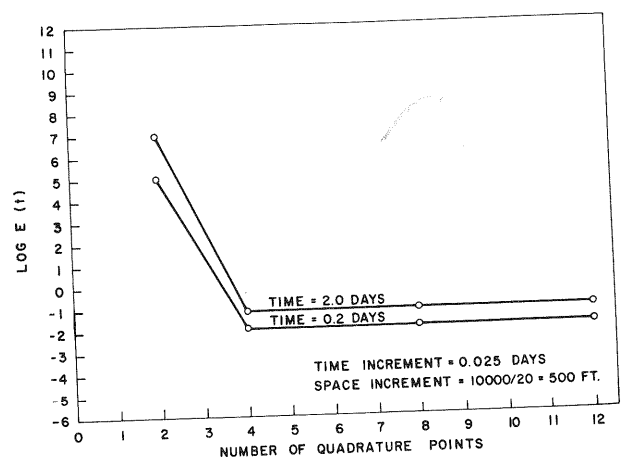

FIG. 2 — EFFECT OF ORDER OF QUADRATURE ON ERROR — BASIC GALERKIN METHOD USING SMOOTH CUBIC FUNCTIONS.

$$E(t) = K_1(t)(\Delta x)^a + K_2(t)(\Delta t)^b + \epsilon(t, \Delta x, \Delta t), \quad \ldots\ldots\ldots\ldots\ldots\ldots (29)$$

where ϵ $(t, \Delta x, \Delta t)$ represents higher order terms. Now, if both Δx and Δt are sufficiently small and Δt is much smaller than Δx so that the terms $K_2(t)(\Delta t)^b$ and ϵ are negligible, then $E(t)$ is given by

$$E(t) \approx K_1(t)(\Delta x)^a, \quad \ldots\ldots\ldots\ldots (30)$$

and a log-log plot of $E(t)$ vs Δx should result in a straight line with slope a, the numerical order of convergence of the method. This slope was determined for each method of solution. The Crank-Nicolson time approximation was used with each of these methods. Fig. 6 presents log $E(t)$ vs log Δx for each method at a time level of 1.0 days. Fig. 7 is a similar plot but at the 2.0-day time level.

In comparing the various methods from the standpoint of computer time needed to obtain a given accuracy, both space and time increments were taken large enough so that the error $E(t)$ was influenced by both time and spatial truncation errors. The comparison was carried out by partitioning the system into 100 mesh blocks for methods believed to be second-order correct in

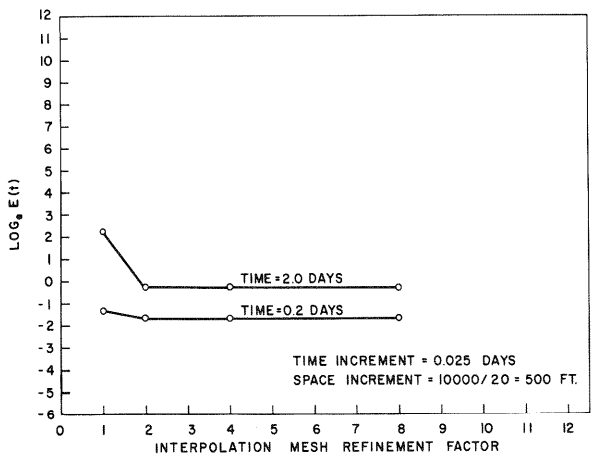

FIG. 3 — EFFECT OF INTERPOLATION MESH REFINEMENT ON ERROR — MODIFIED GALERKIN — CUBIC SPLINE METHOD EMPLOYING CUBIC SPLINE INTERPOLATION.

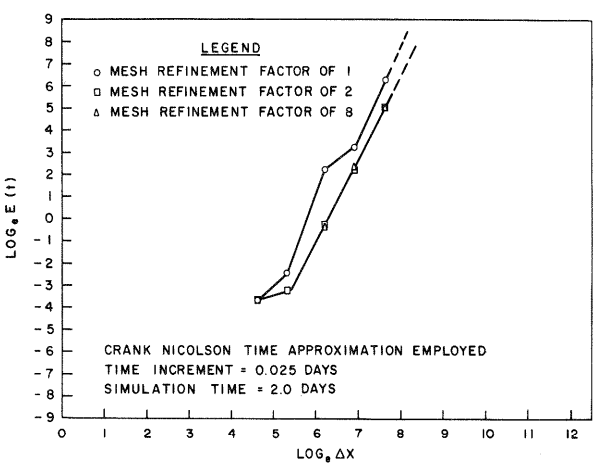

FIG. 4 — EFFECT OF MESH SPACING AND INTERPOLATION MESH REFINEMENT ON ERROR — MODIFIED GALERKIN — CUBIC SPLINE METHOD USING CUBIC SPLINE INTERPOLATION.

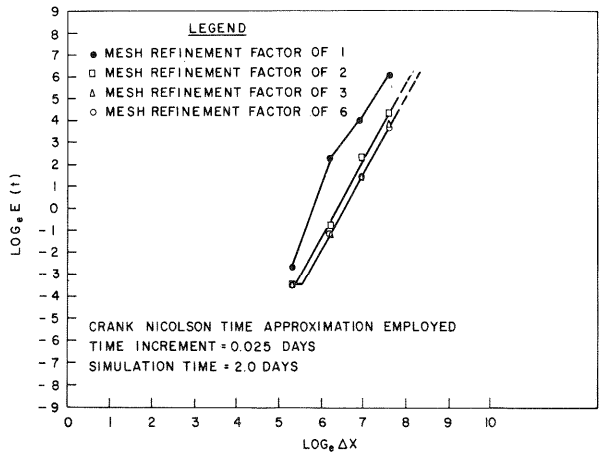

FIG. 5 — EFFECT OF MESH SPACING AND INTERPOLATION MESH REFINEMENT ON ERROR — MODIFIED GALERKIN — SMOOTH CUBIC METHOD USING CUBIC HERMITE INTERPOLATION.

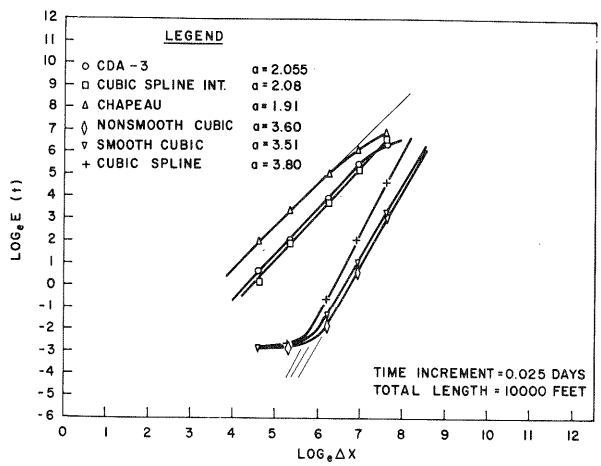

FIG. 6 — EFFECT OF MESH SPACING ON ERROR AT 1 DAY.

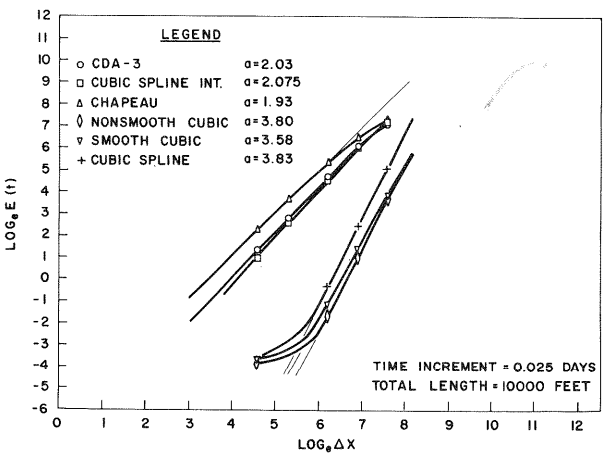

FIG. 7 — EFFECT OF MESH SPACING ON ERROR AT 2 DAYS.

space and into 10 mesh blocks for techniques believed to be fourth-order correct in space. On this basis, the spatial error of all the methods was of the same order of magnitude (i.e., 5.0 to 12.0 psi). After fixing the mesh spacing in this manner, an error curve of log $E(t)$ vs log Δt was prepared for each method. An example of these curves is given in Fig. 9. The combination of six spatial approximations and four time approximations necessitated the preparation of 24 error charts plus additional graphs for each modification of the basic methods. Total computing times were generated for each combination of space and time approximations by selecting a time step corresponding to a given error and calculating the total simulation time from accurate timing data prepared for each method. Table 1 presents normalized computing time (i.e., normalized by dividing all times by the smallest value) for all the methods at a time level of 16.0 days and for log $E(t)$ values of 3.0, 4.0 and 5.0. These log $E(t)$ values represent errors of approximately 20 psi, 55 psi and 140 psi from the true solution or relative errors of approximately 1.0, 2.5 and 7.5 percent, respectively. Fig. 8, prepared from the analytic solution, illustrates the pressure distribution at 16.0 days.

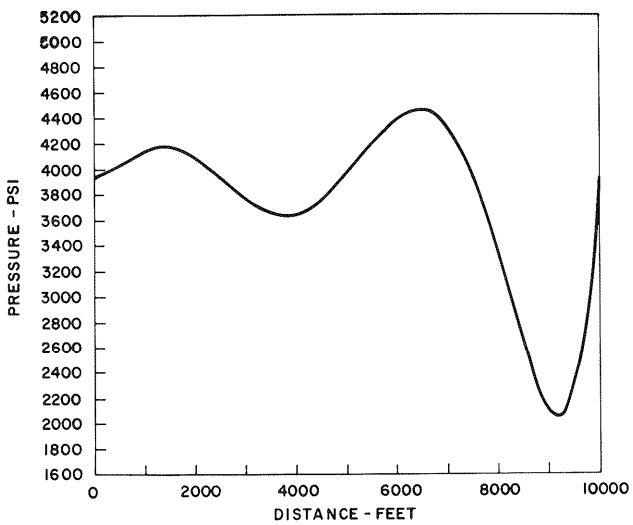

FIG. 8 — PRESSURE DISTRIBUTION AT 16.0 DAYS.

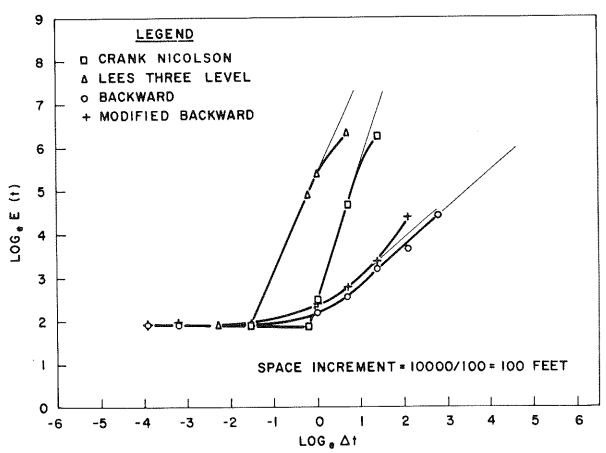

FIG. 9— EFFECT OF TIME-STEP SIZE ON ERROR AT 16.0 DAYS — CDA METHOD OF SOLUTION.

The Galerkin methods represented in Table 1 refer to the formulation defined by Eqs. 12 through 14, where the integrals were evaluated using a consistent quadrature scheme. Table 2 compares normalized computer time (normalized by using the smallest CDA time as the normalizing factor) for the CDA methods with the modified Galerkin-cubic spline and the modified Galerkin-smooth cubic methods employing various interpolation schemes. Table 3 presents normalized computer time for the CDA method and the modified Galerkin-cubic spline method employing a consistent cubic spline interpolation routine for a log $E(t)$ value of −2.0. This error corresponds to a relative error of less than 0.005 percent. To reduce the error to this magnitude, the system was partitioned into 1,600 mesh blocks for the CDA method and 40 blocks for the Galerkin-cubic spline method.

An additional comparison, involving computer core storage, was prepared for most of the methods tested. This comparison is presented in Table 5 and is a normalized comparison, with the program requiring the least amount of storage serving as a reference. As in the computing time comparison, the methods thought to be second-order correct in space were allotted more storage for the main storage arrays than for the methods thought to be fourth-order correct. In Table 5, second-order correct methods were allocated storage to handle 400 mesh

TABLE 1 — NORMALIZED COMPARISON OF COMPUTER TIME FOR SIMULATION TO 16.0 DAYS*

Method of Solution	Accuracy: Log $E = 3.0$	Log $E = 4.0$	Log $E = 5.0$
CN – Chapeau (100)	30.92	55.11	91.95
L3L – Chapeau (100)	55.00	92.00	143.44
BKWD – Chapeau (100)	3.36	4.96	7.17
MBKWD – Chapeau (100)	3.12	3.97	7.97
CN – smooth cubics (10)	16.77	33.52	61.33
L3L – smooth cubics (10)	14.83	30.82	54.05
BKWD – smooth cubics (10)	10.29	5.55	4.00
MBKWD – smooth cubics (10)	2.40	3.28	4.76
CN – nonsmooth cubics (10)	13.77	28.07	51.37
L3L – nonsmooth cubics (10)	11.48	24.93	46.46
BKWD – nonsmooth cubics (10)	6.93	5.64	4.35
MBKWD – nonsmooth cubics (10)	1.87	2.76	3.98
CN – cubic spline (10)	14.39	30.22	53.66
L3L – cubic spline (10)	15.85	24.82	54.02
BKWD – cubic spline (10)	6.85	4.08	3.41
MBKWD – cubic spline (10)	1.65	2.47	3.34
CN – non-Galerkin cubic spline (100)	6.20	11.79	20.91
L3L – non-Galerkin cubic spline (100)	20.12	36.22	61.50
BKWD–non-Galerkin cubic spline (100)	1.10	1.74	3.20
MBKWD–non-Galerkin cubic spline (100)	1.11	1.95	3.17
CN – CDA (100)	3.45	6.59	12.09
L3L – CDA (100)	8.89	15.24	24.26
BKWD – CDA (100)	1.11	1.00	1.00
MBKWD – CDA (100)	1.00	1.36	1.81

*The CN and BKWD schemes employed three iterations per time step.

TABLE 2 — NORMALIZED COMPARISON OF COMPUTER TIME FOR SIMULATION TO 16.0 DAYS*

Method of Solution	Accuracy Log $E = 3.0$	Log $E = 4.0$	Log $E = 5.0$
CN – CDA (100)	3.45	6.59	12.09
BKWD – CDA (100)	1.11	1.00	1.00
MBKWD – CDA (100)	1.00	1.36	1.81
CN – cubic spline (10)	14.39	30.22	53.66
BKWD – cubic spline (10)	6.85	4.08	3.41
MBKWD – cubic spline (10)	1.65	2.47	3.34
CN – cubic spline (10)[1]	7.33	14.36	25.24
BKWD – cubic spline (10)[1]	2.85	1.70	1.78
MBKWD – cubic spline (10)[1]	0.82	1.33	2.14
CN – cubic spline (10)[2]	3.44**	8.28	13.04
BKWD – cubic spline (10)[2]	2.15**	0.96	0.93
MBKWD – cubic spline (10)[2]	0.99**	0.77	0.96
CN – cubic spline (10)[3]	4.38	9.10	16.99
BKWD – cubic spline (10)[3]	1.94	1.16	1.04
MBKWD – cubic spline (10)[3]	0.68	0.86	1.29
CN – smooth cubics (10)[4]	5.64	10.95	15.76
BKWD – smooth cubics (10)[4]	–	1.83	2.25
MBKWD – smooth cubics (10)[4]	–	0.93	1.28
CN – smooth cubics (10)[5]	5.62	11.46	21.62
BKWD – smooth cubics (10)[5]	3.04	1.53	1.24
MBKWD – smooth cubics (10)[5]	0.92	1.20	1.69

[1]Uses consistent integral least-squares interpolation by piecewise cubic spline functions.

[2]Uses consistent (an interpolation mesh twice as fine as the solution mesh) cubic spline interpolation.

[3]Uses cubic spline interpolation employing an interpolation mesh three times as fine as solution mesh.

[4]Uses cubic Hermite interpolation employing an interpolation mesh twice as fine as the solution mesh.

[5]Uses cubic Hermite interpolation employing an interpolation mesh three times as fine as the solution mesh.

*The CN and BKWD schemes employed three-iteration time step.

**A localized refined mesh was employed at the end points for estimating derivatives in the cubic spline interpolation schemes.

TABLE 3 — NORMALIZED COMPARISON OF COMPUTER TIME FOR SIMULATION TO 16.0 DAYS*

Method of Solution	Accuracy Log $E = -2.0$
CN – CDA (1,600)	1.00
BKWD – CDA (1,600)	19.82
MBKWD – CDA (1,600)	7.25
CN – cubic spline (40)**	0.19
BKWD – cubic spline (40)**	3.39
MBKWD – cubic spline (40)**	0.74

*The CN and BKWD schemes employed three iterations per time step.

**Uses consistent (an interpolation mesh twice as fine as the solution mesh) cubic spline interpolation.

TABLE 4 — COMPARISON OF EXPERIMENTAL AND ANTICIPATED THEORETICAL ORDERS OF ACCURACY

Method of Solution*	Experimental	Anticipated Theoretical Values
CDA	2.03	2.0
Non-Galerkin cubic spline	2.08	2.0
Chapeau	1.93	2.0
Smooth cubic	3.58	4.0
Nonsmooth cubic	3.80	4.0
Cubic spline	3.83	4.0

*Tests were carried out using a Crank-Nicolson time approximation with Δt sufficiently small so that the error is dominated by the spatial truncation term.

blocks, whereas fourth-order correct methods were allocated only enough storage to handle 20 mesh blocks. A comparison on this basis is believed to be more realistic since each method will be capable of obtaining a solution with approximately the same magnitude of spatial error.

DISCUSSION

In this section, the numerical results that were outlined in the previous section are discussed. The requirements needed to insure consistent quadrature and interpolation schemes in Galerkin-type problems are outlined. The numerical order of accuracy (convergence) of each of the methods studied is presented, and all methods are compared from the standpoint of computer time required to obtain a given accuracy for a specific problem. Finally, all methods are compared on a computer core storage basis.

CONSISTENT QUADRATURE AND INTERPOLATION SCHEMES

As defined previously, a consistent quadrature scheme is one that preserves the spatial order of convergence of a particular method (i.e., set of basis elements) of solution as well as minimizing, to machine accuracy, the coefficient of $(\Delta x)^a$ in Eq. 30. Figs. 1 and 2 present data on how the error, defined by Eq. 27, behaves when the source vector and coefficient matrices, as defined by Eqs. 12

TABLE 5 — NORMALIZED COMPARISON OF COMPUTER CORE STORAGE REQUIREMENTS

Method of Solution	Normalized Storage
CN – Chapeau (400)	2.656
L3L – Chapeau (400)	2.670
BKWD – Chapeau (400)	2.654
MBKWD – Chapeau (400)	2.593
CN – smooth cubic (20)	1.125
L3L – smooth cubic (20)	1.156
BKWD – smooth cubic (20)	1.102
MBKWD – smooth cubic (20)	1.097
CN – nonsmooth cubic (20)	1.167
L3L – nonsmooth cubic (20)	1.200
BKWD – nonsmooth cubic (20)	1.165
MBKWD – nonsmooth cubic (20)	1.155
CN – cubic spline (20)	1.006
L3L – cubic spline (20)	1.036
BKWD – cubic spline (20)	1.004
MBKWD – cubic spline (20)	1.000
CN – non-Galerkin cubic spline (400)	1.842
L3L – non-Galerkin cubic spline (400)	2.041
BKWD – non-Galerkin cubic spline (400)	1.703
MBKWD – non-Galerkin cubic spline (400)	1.641
CN – CDA (400)	1.512
L3L – CDA (400)	1.549
BKWD – CDA (400)	1.511
MBKWD – CDA (400)	1.449
CN – cubic spline (20)*	1.335
BKWD – cubic spline (20)*	1.333
MBKWD – cubic spline (20)*	1.329
CN – cubic spline (20)**	1.399
BKWD – cubic spline (20)**	1.398
MBKWD – cubic spline (20)**	1.394

*Uses consistent integral least-square interpolation by piecewise cubic spline function.

**Uses consistent cubic spline interpolation.

through 14, are generated using Gaussian quadrature formulas of various orders of accuracy. Fig. 1 presents data for methods using Chapeau-basis functions and Fig. 2 data for methods using smooth cubic-basis functions. Examination of Fig. 1 indicates that the error remains essentially the same for all orders of quadratures employed, the order of the quadratures varying from two-point to twelve-point formulas. Thus, it can be concluded that if a method uses Chapeau-basis elements, a two-point Gaussian quadrature can be considered a consistent quadrature. Analogously, Fig. 2 indicates that a four-point quadrature scheme is necessary in order to prevent the error originating in the quadratures from affecting the accuracy of the method of solution.

Examination of the definition of the elements of the coefficient matrices given in Eqs. 13 and 14 indicates that, in the case of Chapeau functions, the integrals involve polynomials of Degree Two multiplied by a positive weight function. In the case of smooth cubics, the integrals involve polynomials of Degree Six multiplied by the same weight function, i.e., $\beta[p(x,t)]$. Since an n-point Gaussian quadrature scheme exactly integrates polynomials of degree $2n-1$ or less, the results of this study indicate that a quadrature routine that integrates polynomials one degree higher than the degree obtained from the product of the basis functions can be classified as consistent.

Consistent interpolation, as related to the modified Galerkin approach, defined by Eqs. 15 through 17, has been previously defined. The data presented in Fig. 3 discloses that if cubic spline interpolation is employed a mesh refinement factor of two is required in order to prevent the error originating in the interpolation portion of the calculations from affecting the accuracy of the modified Galerkin-cubic spline method of solution. Consistency tests carried out using integral least-squares interpolation of the coefficients by continuous piecewise cubic splines coupled with the Galerkin-cubic spline method of solution resulted in data similar to that presented in Fig. 2. This method of interpolation requires the approximation of integrals, and these tests confirmed the results of the consistent quadrature tests in that a four-point Gaussian quadrature scheme is necessary for consistent interpolation.

Figs. 4 and 5 present additional data that can be used to illustrate the meaning of consistent interpolation. The data in these two figures clearly indicates the reduction in the magnitude of the coefficient $K_1(t)$ of Eq. 30 as the mesh refinement factor increases (i.e., as the interpolation mesh becomes finer). Fig. 4, which presents data on the modified Galerkin-cubic spline method employing cubic spline interpolation, demonstrates that no measurable reduction in error can be accomplished by employing a mesh refinement factor greater than two. Similarly, Fig. 5, which presents data on the modified Galerkin-smooth cubic method employing smooth cubic Hermite interpolation, illustrates that a mesh refinement factor between two and three is sufficient to insure a consistent interpolation scheme for this method of solution. It should also be noted that in both figures a mesh refinement factor of one results in local changes in the order of convergence (the general trend of these curves is approximately the same as the curves generated using a larger-mesh refinement factor) even though both interpolation schemes are fourth-order correct—the same order as the method of solution.

In summary, although no theoretical proof exists at the present time, it is logical to conclude that any interpolation scheme that has the same (or a higher) order of convergence as the over-all method of solution should preserve the value of the exponent a in Eq. 30 (i.e., be consistent in the sense of Herbold's[5] definition). In addition, as the data just presented and discussed indicate, it is possible, by using interpolation schemes with the same order of accuracy as the method of solution coupled with the concept of a refined interpolation mesh, not only to maintain the order of convergence of the over-all method of solution (i.e., preserve exponent a) but also to minimize, to machine accuracy, the coefficient $K_1(t)$ of Eq. 30. Thus, since the modified Galerkin method, employing interpolation methods based on a partition finer than the solution partition, is economically attractive (to be demonstrated later) and since it requires little additional programming effort, it can be concluded that the concept of consistent interpolation as defined in this paper is an efficient procedure for keeping the errors associated with the Galerkin technique to a minimum.

ORDER OF THE APPROXIMATIONS

As outlined previously, the numerical order of convergence of each method was determined by plotting log $E(t)$ vs log Δx. This type of data is used to distinguish the higher-order methods from lower-order methods. Figs. 6 and 7 present these data for time levels of 1.0 and 2.0 days, respectively. Departure from straight lines for large value of Δx can be attributed to the term $\epsilon(t, \Delta x, \Delta t)$ in Eq. 29. For the large values of Δx, its contributions to the total error term is not negligible. Deviation from the straight line for small values of Δx can be attributed to time truncation error. In this portion of the curve, the term $K_2(t)(\Delta t)^b$ is not small compared to $K_1(t)(\Delta x)^a$. Also note that the curves exhibiting the highest values for the slope a approach a constant for small Δx; this is the time truncation error.

Table 4 compares a, the numerical order of convergence of each method, with anticipated theoretical values. Anticipated or expected values are used here since no theoretical rate of convergence data exists for nonlinear parabolic problems of the type studied here. One exception to this is a recent publication[13] that presents theoretical convergence data for Galerkin methods as applied to nonlinear parabolic equations. However, these results indicate a lower-order convergence rate than can be obtained

experimentally. The three methods, Chapeau, CDA, and the non-Galerkin cubic spline interpolation method, have slopes that agree very well with expected theoretical values. Of the three second-order methods, Figs. 6 and 7 indicate that the non-Galerkin cubic spline interpolation method is slightly better than CDA and that both are superior to the Chapeau approximation. Although all three have approximately the same order of convergence, the factor $K_1(t)$ in Eq. 30 is smallest for the non-Galerkin cubic spline interpolation method. Comparison of the three higher-order methods indicates that the experimentally determined order of convergence does not agree as closely with the anticipated theoretical value as the second-order methods. Additional computational results using an error defined by the L_2 norm improved the agreement between the numerical and the anticipated rate of convergence for the Galerkin-cubic spline method. The data in Figs. 6 and 7 indicate that the nonsmooth cubic method is slightly superior to the other two methods since the coefficient of $(\Delta x)^a$ is smallest for this method.

In summary, the most important information conveyed by the data in Table 4 is not the closeness of agreement between numerical order of convergence data and expected theoretical values, but simply the distinction between high- and lower-order accuracy methods. The ability to distinguish the order of convergence is important because the higher-order methods require a partition with substantially fewer mesh blocks than lower-order methods for numerical solutions of a specified accuracy.

COMPUTING TIME AND STORAGE COMPARISON

Although knowledge of the order of convergence is important in selecting methods, the most important point when comparing different methods is how much computing time is expended to obtain a given accuracy. A second consideration is how much core storage is required for each method. Answers to both of these questions can be obtained by examining Tables 1 through 3 and 5.

Examination of Table 1 data indicates that the standard central finite-difference space approximation coupled with a modified backward-time approximation is the fastest for a relative error less than 1.0 percent (roughly an error of 20 psi), and that for larger errors, approximately 2.5 and 7.0 percent (or 55 psi and 150 psi), the CDA method using a backward-time approximation is superior. Comparing just the Galerkin methods, which refer to the basic formulation represented by Eqs. 12 through 14, the one employing cubic splines with a modified backward-time approximation is the fastest. These results are in direct contrast to results reported for a linear problem.[1] In linear problems, the Galerkin-type methods were far superior to the finite-difference methods. The primary reason for the increase in computing time associated with the Galerkin-type methods can be attributed to the many quadrature operations that must be carried out to generate the coefficient matrices in Eq. 10. In linear problems, these matrices remain unchanged, and the associated quadrature need be performed just once. For nonlinear problems, these matrices necessarily vary with each time step.

The necessity of utilizing the modified Galerkin method, as defined by Eqs. 15 through 17, if the Galerkin approach is to compete with standard numerical techniques is demonstrated by the data of Tables 2 and 3. In these tables, which utilize the CDA time as a normalizing factor, the Galerkin (modified form) is superior. For a maximum error less than 1.0 percent the modified Galerkin-cubic spline method using a modified backward-time approximation and cubic spline interpolation with a mesh refinement factor of three is the fastest. For the other two accuracy values, the modified Galerkin-cubic spline method using a backward-time approximation and cubic spline interpolation with a mesh refinement factor of two is superior. Table 2 indicates that the computing time associated with the basic Galerkin method is reduced by approximately a factor of two when integral least-squares interpolation is used and by a factor slightly less than three when the cubic spline interpolation technique is utilized. A similar reduction also resulted for the basic Galerkin-smooth cubic method when the modified Galerkin-smooth cubic technique using cubic Hermite interpolation was employed. This can be substantiated by comparing the time given for these methods in Tables 1 and 2. It should also be pointed out that it was not possible to obtain numerical solutions with errors log $E = 3.0$ and log $E = 4.0$ using the modified Galerkin method employing cubic and quintic spline interpolation and a mesh refinement factor of one; also for an accuracy of log $E = 5.0$ this approach was not competitive with the Galerkin methods represented in Table 2. In addition, it was also found that runs utilizing a finer solution mesh (i.e., 20 mesh blocks) and a mesh refinement factor of one were not competitive with Table 2 Galerkin methods. The integral least-squares interpolation with two-point quadrature approximation resulted in similar problems. The reason for the uncompetitive status of these approaches was attributed to the use of inconsistent interpolation and quadrature routines. That is, the error introduced by inconsistent interpolation was large enough to result in an unattractive technique.

Additional points of interest illustrated in Tables 1 and 2 concern the different time approximations employed with each space approximation. For each type of space approximation, the MBKWD and BKWD time approximations were superior to the Crank-Nicolson and Lees three-level methods. It was hoped that the Lees three-level time approximation with the coefficients centered at the middle time level would be an efficient scheme since the approximation has a $0[(\Delta t)^2]$ local truncation error and requires no iterations on the nonlinear coefficients to proceed from one time level to the next. However, even though log $E(t)$ vs log Δt did

indicate a second-order scheme, the magnitude of the error was very large compared with the other time approximations. In addition, oscillations in error vs time plots were noticed. Under these conditions, relatively small time steps were required to reduce the errors to the values used in Tables 1 and 2. The surprising point, however, concerning the different time approximations is the superiority of first-order approximations over higher-order time approximations. As pointed out previously, the backward and modified backward time approximations were superior.

The apparently inconsistent results of generating a smaller error by utilizing a first-order correct time approximation instead of a second-order correct scheme for a given time-step size can be explained by examining Fig. 9, which presents log $E(t)$ vs log Δt for the CDA space approximation at a simulation time of 16.0 days. Inspection of this figure indicates that to obtain a solution with an error of log $E = 4.0$ using the CN approximation requires a time step approximately six times smaller than the time step required for the BKWD technique. This point illustrates, for example, why the CN technique, which is second-order correct, requires approximately six times as much work as the BKWD method, which is first-order correct, even though both methods involve approximately the same amount of algebraic manipulations (see Table 1 log $E = 4.0$ data).

Further examination of Fig. 9 indicates that, if a smaller mesh size had been used, the curves representing first- and second-order correct methods would have crossed (i.e., extend the linear portion of each curve in the direction of smaller Δt) and for a sufficiently small Δt second-order methods would indeed provide a smaller error than first-order methods. This point can be verified by examining Table 3, which presents normalized timing data for the modified Galerkin-cubic spline and CDA methods for a log $E(t)$ value of -2.0 (i.e., a relative error in the numerical solution of less than 0.005 percent). In this table the second-order CN method is superior to both the BKWD and MBKWD approximation. The CN approximation is superior because the portion of the log $E(t)$ vs log Δt plot, which is used to determine the time step needed to generate numerical solutions with a log $E = -2.0$ error, is in the region where second-order methods exhibit smaller errors than first-order methods for the same size of time step. In this particular range of accuracy, the L3L time approximation would likely be competitive with the CN method. However, since the spatial error is of the order of 6.0 psi (i.e., a relative error less than 0.5 percent) in the data presented in Tables 1 and 2, the practical area of interest of the test problem lies in the region where first-order correct time approximations are superior to second-order methods.

On an over-all basis, the data indicate that the MBKWD and BKWD-modified Galerkin-cubic spline methods are superior to the others for a practical range of accuracy and the particular problem chosen. The data of Table 3 also indicate that, as the accuracy requirements become more stringent, the Galerkin methods become more attractive. In addition, when highly accurate numerical solutions are needed, the second-order time approximations would be definitely superior to first-order methods.

In comparing the two spatial approximations, CDA and non-Galerkin cubic spline interpolation, the latter approach has the same advantage as the Galerkin method, namely it generates a continuous solution vs the discrete values obtained by the conventional finite-difference techniques. In addition, the solution provided by the non-Galerkin method is twice continuously differentiable. However, it has recently been proved by Swartz and Varga[12] that it is possible to interpolate second-order correct discrete values, such as those generated by the CDA space approximation at a fixed time level, with continuous, piecewise cubic spline functions and obtain an approximation that is globally second-order correct. Thus, a method employing cubic spline interpolation of discrete CDA solution values would be superior to the non-Galerkin approach of Albasiny because the interpolation portion of the problem would only be carried out at simultation times of interest to the engineer. In Albasiny's approach, the formulation is such that the discrete solution and the interpolation are carried out simultaneously at every time step. Thus, in conclusion, it appears that the CDA approximation coupled with cubic spline interpolation of the discrete solution values generated by this method is a more flexible and efficient approach to solving the subject problem than the modified approach of Albasiny.

The core-storage requirement comparison, presented in Table 5 on the basis of having spatial truncation errors of the same order of magnitude for all methods, indicates that the basic Galerkin-type method employing cubic spline basis elements in conjunction with the MBKWD time approximation requires the least amount of storage. This data also shows that the modified-Galerkin approach also requires less core storage than the standard CDA method.

SUMMARY AND CONCLUSIONS

Use of Galerkin-type methods employing continuous piecewise polynomial functions as basis elements to solve nonlinear, time-dependent two-point boundary value problems has been demonstrated. A modification to the basic Galerkin technique for nonlinear problems was developed and is examined in detail. The necessity of using this modification in order to obtain an efficient solution method is also established.

In addition to the Galerkin methods, a method which uses a non-Galerkin cubic spline interpolation procedure was introduced. These methods were tested extensively and compared with results obtained from the more conventional finite-difference approach to solving nonlinear problems. Four different time approximations were evaluated. The

tests were carried out on the equation describing the transient flow of a real gas in a porous medium. Results of these tests, which are restricted to equations of the type studied, indicated that:

1. Criteria for consistent interpolation and quadrature operations were determined. Numerical experimentation indicates that a quadrature routine that integrates polynomials one degree higher than the degree obtained from the product of two basis functions used to represent the solution of a problem can be classified as consistent. Numerical work also defined the conditions needed to insure consistent interpolation methods. Consistent cubic spline interpolation, in connection with the modified Galerkin-cubic spline method of solution, requires an interpolation mesh twice as fine as the solution mesh (i.e., a mesh refinement factor of two). Consistent cubic Hermite interpolation, in connection with the modified-Galerkin smooth cubic method of solution, requires an interpolation mesh three times as fine as the solution mesh (i.e., a mesh refinement factor of three).

2. The numerical order of accuracy of each of the methods tested in this study was determined. These are presented in Table 4.

3. The degree of accuracy required in the numerical solution determines which time approximation is superior on the basis of computer time expended to obtain a given accuracy. For numerical solutions with relative errors varying from approximately 1.0 to 7.5 percent, the first-order correct time approximations were superior. The second-order CN method was superior for numerical solutions with relative errors less than 0.005 percent.

4. The modified-Galerkin-cubic spline method employing consistent cubic spline interpolation coupled the MBKWD and BKWD time approximations were superior in regard to computer time expended for a given accuracy.

5. Use of consistent interpolation and quadrature schemes, as opposed to inconsistent schemes, provided the most economical approach to solving the test problem with the Galerkin method.

6. On the basis of generating a solution with approximately the same magnitude of spatial truncation error, the Galerkin-cubic spline method using the modified backward-time approximation requires the least amount of computer core storage.

7. Of the four types of basis functions used in the Galerkin approach, the cubic spline in conjunction with a modified backward-time approximation was the most efficient in regard to both storage required and computational effort to obtain a given accuracy.

8. The CDA spatial approximation coupled with cubic spline interpolation of the discrete solution values generated by this method would be a more efficient procedure than the modified Albasiny approach.

Although no mention of problems involving more than one dimension was made in the previous material, it has already been pointed out[1] that the savings in computer time for one-dimensional problems would be essentially squared when going to two dimensions. Galerkin two-dimensional formulations for certain problems have been outlined by others.[4,13,14]

As was stated in the introductory statements of this section, the previously stated conclusions are restricted to equations of the type studied. However, it is believed the following more general points can be deduced from the results of this study.

1. The data presented on the various time approximations indicates that just because a time approximation is second-order correct it does not necessarily follow that it is superior to a first-order method. Asymptotically (i.e., as $\Delta t \to 0$) second-order methods are superior to first-order methods, but as this work demonstrates, conditions exist where first-order methods are superior to second-order methods. However, when comparing second- and first-order methods, it must also be pointed out that the second-order method will result in a correspondingly larger reduction in the error for a given time-step size reduction. This is, of course, independent of the accuracy range considered.

2. If the Galerkin method is to be utilized in an economic manner (i.e., from the standpoint of computer time expended to obtain a given accuracy), then the modified version as defined by Eqs. 15 through 17 should be utilized.

3. As this paper demonstrates, the concept of consistent quadrature and interpolation schemes as used in the Galerkin formulation are important considerations. It appears the most efficient use of the Galerkin methods involves defining and using consistent schemes in the solution of nonlinear problems.

Finally, the following material presents some general observations concerning the Galerkin methods. First, as the results of this study point out, a consistent interpolation scheme can be obtained by choosing an interpolation technique with the same (or higher) order of convergence as the method of solution and employing a sufficiently fine interpolation mesh to minimize the errors introduced by the interpolation procedure. Since the appropriate mesh refinement factor will probably vary from problem to problem and with the choice of basis functions (see, for example, the smooth cubic and cubic spline methods reported previously), it will be necessary to determine it for each particular case. The mesh refinement factor needed for consistent interpolation can be easily determined by making a few test runs (each test using a different interpolation mesh) and comparing successive solution values. Second, in regard to the computational superiority exhibited by the method employing cubic spline functions, the following precautionary comments should be noted. In problems involving a single phase, such as the gas flow problem studied here, it appears likely that the cubic spline basis elements would be the optimum choice. However, for multiphase flow problems or miscible flow problems, both of which can involve sharp changes in a dependent variable, other basis

elements may provide a better approximation to the solution.

NOMENCLATURE*

a = order of convergence of spatial approximations

A = conversion constant, $A = 158.07$

b = order of convergence of time approximations

$\overline{B}$ = conversion constant, $\overline{B} = 1{,}696.41$

c_i = time-dependent coefficients in the Galerkin formulation

E = maximum error defined by absolute value of the difference between the numerical solution and analytic solution, psi

k = absolute permeability, md

L = total system length, ft

M = molecular weight, lb_m/lb-mole

p = pressure, psi

$\hat{p}$ = pressure determined by numerical methods, psi

p_o = initial pressure, psi

S = volumetric source term, lb_m/cu ft-day

$\overline{S}$ = denotes cubic spline polynomial in non-Galerkin cubic spline method

t = time, days

T = temperature, °R

w_i = ith basis element in Galerkin formulation

x = rectangular coordinate, ft

z = gas deviation factor, dimensionless

α = differential equation coefficient, Eq. 1

α_1 = solution constant, $\alpha_1 = 0.20$

α_2 = solution constant, $\alpha_2 = 140.0$

α_3 = solution constant, $\alpha_3 = 40.0$

β = differential equation coefficient, Eq. 1

γ = constant $\left(\gamma = \dfrac{k_{\max} p_o}{\phi \mu_o L^2} \text{ day}^{-1}\right)$

ϕ = porosity, dimensionless

μ = viscosity, cp

MATHEMATICAL SYMBOLS

$C^1[0,L]$ = continuously differentiable functions on $[0,L]$

Δ = represents an incremental unit of space or time; also used to represent a partition of a finite interval

log = denotes natural logarithm

$\langle , \rangle$ = L_2 inner product defined on $[0,L]$ as

$$\langle f,g \rangle \equiv \int_0^L f(x)\, g(x)\, dx$$

w' = denotes derivative operation $w' = \dfrac{dw}{dx}$ and $w'' = \dfrac{d^2w}{dx^2}$

*Symbols underlined in this paper represent vectors.

ACKNOWLEDGMENT

The authors would like to thank the management of Gulf Research & Development Co. for permission to publish this paper.

REFERENCES

1. Price, H. S., Cavendish, J. C. and Varga, R. S.: "Numerical Methods of Higher-Order Accuracy for Diffusion-Convection Equations", *Soc. Pet. Eng. J.* (Sept., 1968) 293-303.
2. Price, H. S. and Varga, R. S.: "Error Bounds for Semidiscrete Galerkin Approximations of Parabolic Problems with Applications to Petroleum Reservoir Mechanics", *Numerical Solution of Field Problems in Continuum Physics*, G. Birkoff and R. Varga, Ed., *SIAM-AMS Proc.*, American Mathematical Society, Providence, R. I. (1970) Vol. 2, 74-94.
3. Varga, R. S.: *Matrix Iterative Analysis*, Prentice-Hall, Inc., Englewood Cliffs, N. J. (1962).
4. Cavendish, J. C., Price, H. S. and Varga, R. S.: "Galerkin Methods for the Numerical Solution of Boundary Value Problems", *Soc. Pet. Eng. J.* (June, 1969) 204-220.
5. Herbold, R. J.: "Consistent Quadrature Schemes for the Numerical Solution of Boundary Value Problems by Variational Techniques", PhD dissertation, Case-Western Reserve U., Cleveland (June, 1968).
6. Varga, R. S.: "Hermite Interpolation-Type Ritz Methods for Two-Point Boundary Value Problems", *Numerical Solution of Partial Differential Equations*, J. H. Bramble, Ed., Academic Press, Inc., New York (1966) 365-373.
7. Ciarlet, P. G., Schultz, M. H. and Varga, R. S.: "Numerical Methods of Higher Order Accuracy for Nonlinear Boundary Value Problems: I. One-Dimensional Problems", *Numerische Mathematik* (1967) Vol. 9, 394-430.
8. Albasiny, E. L. and Hoskins, W. D.: "Cubic Spline Solutions to Two-Point Boundary Value Problems", *The Computer J.* (May, 1969) Vol. 12, No. 2, 151-153.
9. Ahlberg, J. H., Nilson, E. N. and Walsh, J. L.: "The Theory of Splines and Their Application", Academic Press, Inc., New York (1967).
10. *Modern Computing Methods*, Second ed., National Physical Laboratory, Her Majesty's Stationary Office, London (1961).
11. Lees, Milton: "A Linear Three-Level Difference Scheme for Quasilinear Parabolic Equations", *Math. Computations* (Oct., 1966) 516-522.
12. Swartz, Blair K. and Varga, R. S.: "Error Bounds for Spline and Spline Interpolation", *J. Approx. Theory*, to be published.
13. Douglas, Jim, Jr. and Dupont, Todd: "Galerkin Methods for Parabolic Equations", *J. on Numerical Analysis*, SIAM (Dec., 1970) Vol. 7, No. 4, 575-626.
14. Jennings, J. W.: "The Application of Variational Methods for Calculating Two-Dimensional Immiscible Displacement in Porous Media", PhD dissertation, U. of Pittsburgh (1969).
15. Douglas, Jim, Jr., Dupont, Todd and Rachford, H. H., Jr.: "The Application of Variational Methods to Waterflooding Problems", *Cdn. J. of Pet. Tech.* (July-Sept., 1969) 79-85.

APPENDIX

NON-GALERKIN CUBIC SPLINE INTERPOLATION METHOD

Albasiny and Hoskins[8] show that the cubic spline approximation to a two-point boundary value problem for the ordinary differential equation

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} + g(x)y = r(x) \quad \ldots \quad \text{(A-1)}$$

reduces to the solution of a three-term recurrence relationship. The actual mechanics of this method will be illustrated by employing a simple example furnished by Albasiny.

Using Eq. 20, the cubic spline interpolation to $y(x)$ at the grid points x_i, $i = 0, 1, \ldots, N$, where $x_i = x_o + ih$, and continuity of the first derivative at the grid point results in the following relationship between M_i and $y(x_i)$. (See Ref. 9.)

$$\frac{h}{6}M_{i-1} + \frac{2h}{3}M_i + \frac{h}{6}M_{i+1} = \frac{y_{i+1} - 2y_i + y_{i-1}}{h}; \quad i = 1, 2, \ldots N-1. \quad \ldots \quad \text{(A-2)}$$

For the simplified case where the first derivative is absent in Eq. A-1, it follows that

$$M_i = r_i - g_i y_i; \quad i = 0, 1, 2, \ldots N, \quad \ldots \quad \text{(A-3)}$$

and substitution into Eq. A-2 yields

$$\left(1 + \frac{h^2}{6}g_{i-1}\right)y_{i-1} - \left(2 - \frac{2h^2}{3}g_i\right)y_i + \left(1 + \frac{h^2}{6}g_{i+1}\right)y_{i+1} = \frac{h^2}{6}\left(r_{i-1} + 4r_i + r_{i+1}\right)$$
$$i = 1, 2, \ldots N-1 \quad \ldots \quad \text{(A-4)}$$

Thus, Eq. A-4, with the necessary modifications for boundary conditions, represents a tridiagonal set of equations that can be solved for $y_o, y_1, \ldots, y_N$. The complete cubic spline solution is that given by Eq. A-3 and Eq. 20.

Albasiny derives similar equations for problems where the first derivative is present, and this is the case of interest in this paper. Since the equations for this class of problems are quite lengthy and are readily available elsewhere,[8] they will not be presented here. Instead, Eq. 1 will be modified and the necessary correlation between Eq. A-1 and the modified version of Eq. 1 will be outlined.

Modify Eq. 1 by expanding the space-derivative term and discretizing the time derivative using a backward-time approximation. The result is:

$$\left.\frac{d^2p}{dx^2}\right|_{n+1} + \left(\frac{1}{\alpha(p)}\frac{d\alpha}{dp}\frac{dp}{dx}\right)_{n+1}\left.\frac{dp}{dx}\right|_{n+1} - \frac{1}{\Delta t}\left(\frac{\beta(p)}{\alpha(p)}\right)p_{n+1} = -\frac{1}{\Delta t}\left(\frac{\beta(p)}{\alpha(p)}\right)_{n+1}p_n - \frac{\bar{B}T}{M}\left(\frac{1}{\alpha(p)}S(x,t)\right)_{n+1} \quad \ldots \quad \text{(A-5)}$$

A direct comparison of Eqs. A-5 and A-1 provides the following relations:

$$f(x) = \left(\frac{1}{\alpha(p)}\frac{d\alpha}{dp}\frac{dp}{dx}\right)_{n+1} \quad \ldots \quad \text{(A-6)}$$

$$g(x) = -\frac{1}{\Delta t}\left(\frac{\beta(p)}{\alpha(p)}\right)_{n+1} \quad \ldots \quad \text{(A-7)}$$

$$r(x) = g(x)p_n - \frac{\bar{B}T}{M}\left(\frac{1}{\alpha(p)}S(x,t)\right)_{n+1} \quad \ldots \quad \text{(A-8)}$$

Using the above equations and the relations presented by Albasiny, it is possible to arrive at a set of equations for the pressure at the grid points. This pressure is in turn used to compute M_i values, and then both of these are used to generate the complete cubic spline solution. In cases where the resulting algebraic equations are nonlinear, an iteration scheme similar to the one previously outlined is used.

★★★