ON A NEW APPROACH TO THE COMPUTATIONAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

BY RICHARD BELLMAN, ROBERT KALABA, AND BELLA KOTKIN

THE RAND CORPORATION, SANTA MONICA, CALIFORNIA

Communicated by S. Chandrasekhar, May 31, 1962

1. Introduction.—At the present time, there exist several approaches to the numerical solution of partial differential equations, of which the most versatile and frequently employed is that of approximation by means of difference equations. The technique has many advantages: conceptual simplicity, wide applicability, and ready suitability for digital computation. It also has disadvantages: pre-dilection toward instability and frequent requirement for large storage capabilities and excessive computing time, particularly in the treatment of multidimensional equations.

In this note, we wish to indicate a modified approach which may be superior in some situations. It is based upon two themes. The first is that of using a more efficient way of recreating a function than by storing its values at grid points, and the second is the idea that an approximating algorithm should as clearly as possible exhibit the properties of the actual solution. Thus, for example, if the solution is nonnegative, this fact should be evident from the relations used to obtain it computationally. It is to be expected that algorithms with these replicating properties will be more stable than those without these properties. Whether or not algorithms of the desired type always exist is an interesting and unsolved problem at the present time.

2. The Equation \( u_t = uu_x \).—To illustrate these ideas in a simple setting, let us consider the equation

\[
\begin{align*}
u_t &= uu_x, \\
u(x,0) &= g(x),
\end{align*}
\]

which has the great merit of possessing an explicit analytic solution,

\[
u = g(x + ut),
\]

and which displays a "shock" phenomenon. Both of these characteristics are extremely useful for testing computational techniques. Let us assume, for convenience, that \( g(x) \) is an odd function, periodic of period 2.

In place of the usual type of difference approximation, we use the relation

\[
\begin{align*}
u(x,t + \Delta) &= \nu(x + \nu(x,t)\Delta t),
\end{align*}
\]
which clearly preserves boundedness and nonnegativity, and which tends to (2.1) as $\Delta \to 0$. The variable $t$ is constrained to assume the values $0, \Delta, 2\Delta, \ldots$, but $x$ is not constrained in this way. At each stage of the calculation, $u(x,t)$ is obtained by means of the finite sum

$$u(x,t) \cong \sum_{n=1}^{M} u_n(t) \sin n\pi x,$$

(2.4)

where the coefficients $u_n(t)$ are determined by means of the approximate quadrature

$$u_n(t) = 2 \int_{0}^{1} u(x,t) \sin n\pi x \, dx \cong \frac{2}{R} \sum_{k=1}^{R-1} u(k/R,t) \sin (n\pi k/R).$$

(2.5)

Thus, the values $u(k/R,t)$, $k = 1, 2, \ldots, R - 1$, store the function at time $t$. Using these values, we can compute $u(x,t)$ and hence, by way of (2.3), $u(x,t + \Delta)$ for $x = k/R$, $k = 1, 2, \ldots, R$. Of course, Filon’s or other quadrature schemes could be used.

Similar procedure can be used for equations such as

(a) $u_t = vu_x + \phi(u),$

(b) $g(x)u_t = uu_x,$

(c) $u_t = uu_x + vu_y + g(u,v),

v_t = vv_y + h(u,v),$

(d) $u_N(c) = \max_{0 \leq x \leq c} \{ g_N(v) + u_{N-1}(c - v) \}$ (see ref. 2),

(2.6)

and many others.

3. **Discussion of Numerical Results.**—A FORTRAN program for the IBM-7090 was written to test the procedure. We considered the case for which

$$g(x) = 0.1 \sin \pi x$$

(3.1)

and we chose

$$M = R = 10, \quad \Delta = 0.1$$

(3.2)

Execution requires 2.5 min for a history through $t = 10$. For times less than three, we found that the error in $u(x,t)$ did not exceed three parts in one thousand. Furthermore, the Lagrange expansion formula applied to the exact solution of (2.2), in conjunction with equations (3.1) and (2.2), shows that the coefficients of the terms $\sin \pi x, \sin 2\pi x, \text{ and } \sin 3\pi x$ are approximately proportional to $t^3, t^4, \text{ and } t^5$, respectively for small $t$. This behavior of the harmonics was evident when we reduced $\Delta$ to the value 0.01. For $t = 10$, the time at which shock occurs, the values of $u$ at $x = 0.2, 0.4, 0.6, \text{ and } 0.8$ were in error by about 3 per cent. Nearer $x = 0$, the point at which the shock occurs, much greater relative errors occurred, but qualitative behavior indicative of a shock was apparent.

A discussion of the results obtained from the use of these methods on the functional equations of dynamic programming will be presented separately. For an earlier application of (1.3), see reference 1.

4. **Improvement of Approximation.**—The problem of improving the accuracy of (2.3) while preserving its character is an interesting one. Thus, for example,

$$u(x,t + \Delta) = u(x + u(x,t)\Delta,t)\Delta,t$$

(4.1)
is accurate to $O(\Delta^3)$, while (2.3) is accurate only to $O(\Delta^2)$. Interesting problems of this nature arise in all calculations of this type and will be discussed subsequently.


---

**ORBIT DETERMINATION AS A MULTI-POINT BOUNDARY-VALUE PROBLEM AND QUASILINEARIZATION**

**By Richard Bellman, Harriet Kagiwada, and Robert Kalaba**

**The RAND Corporation, Santa Monica, California**

Communicated by S. Chandrasekhar, May 31, 1962

1. **Introduction.**—The basic ideas underlying the solution of boundary-value problems via quasilinearization are discussed in references 1 and 2. Results of some numerical experiments involving two-point boundary-value problems for nonlinear ordinary differential equations (Euler equations) are available in references 2 and 3, while results for nonlinear partial differential equations are contained in reference 4. The purpose of this note is to describe an application to the computational solution of some general $N$-point boundary-value problems and to present an application to orbit determination.

2. **An $N$-Point Boundary-Value Problem.**—Consider an $N$-dimensional vector $x(t) = (x_1(t), x_2(t), \ldots, x_N(t))$ which is a solution of the vector system of equations

$$\dot{x} = f(x, t), \quad 0 \leq t \leq b,$$

and in addition satisfies the $N$ conditions

$$\sum_{j=1}^{N} a_{ij}(t_i)x_j(t_i) = b_i, \quad i = 1, 2, \ldots, N,$$

where

$$0 \leq t_1 \leq t_2 \leq \ldots \leq t_N \leq b.$$

Let us assume that the equations (2.1) and (2.2) possess a unique solution on the interval $[0, b]$. Our objective is to provide an efficient computational algorithm for determining the function $x(t)$ on the interval $[0, b]$.

3. **Solution via Quasilinearization.**—Let $x^{(0)}(t)$ be an initial approximation on the given interval. The $(k+1)$st approximation, $x^{(k+1)}(t)$, is to be obtained from the $k$th approximation, $x^{(k)}(t)$, through use of the relations

$$\dot{x}_i^{(k+1)} = f_i(x^{(k)}, t) + \sum_{j=1}^{N} (x_j^{(k+1)} - x_j^{(k)}) \frac{\partial f_i(x^{(k)}, t)}{\partial x_j},$$

$$i = 1, 2, \ldots, N \quad \text{and} \quad k = 0, 1, 2, \ldots, \quad (3.1)$$

with the boundary conditions