JOURNAL OF COMPUTATIONAL PHYSICS **143,** 291–311 (1998)
ARTICLE NO. CP985987

Solving Integral Equations by Reconstruction in Isomorphic Taylor Coefficient Spaces

H. C. Motin

849 East Elizabeth Street, Pasadena, California 91104-2346
E-mail: hmotin@ibm.net

Received December 18, 1997; revised March 24, 1998

A new method for numerically solving integral equations is presented. Several examples are shown to demonstrate the accuracy of the procedure. Using the procedure, one can in principle solve any properly constructed integral equation for solutions in the analytic domain. © 1998 Academic Press

1. INTRODUCTION

A new procedure now exists for numerically solving integral equations. It works by transforming each equation system into an equivalent one of mathematically linked Taylor series coefficients. The procedure is viable because, using it, one can in principle solve any properly posed integral equation system (for solutions in the analytic domain, only). Also, it produces numerical results that are in some cases more accurate than from many other techniques. The procedure can be implemented using software that operates similar to a modern spreadsheet.

Strictly speaking, using Taylor series expansions to solve integral or differential equations is not new. Today, however, they are rarely used as the sole or primary solution vehicle, because of inherent difficulties involving these expansions. For example, in general they have poor radius of convergence qualities. To get a sizeable radius, suitable for solving most physical problems of interest, each expansion usually requires a large number of terms. Constructing them is most often time consuming and usually error prone, because it requires continuous partial differentiation. Finally, storing and manipulating large expansions in a solution algorithm is often technically challenging.

In contrast, our procedure does not simply or exclusively use Taylor expansions. Instead, it uses Taylor series components and in a specialized and adapted way. It is this specialized adaptation that is new. It allows us to overcome the difficulties associated with Taylor expansions.

0021-9991/98 $25.00

This paper is divided into four sections. In Section 2 we describe the integral solution procedure in some detail, using illustrative examples and equations. We illustrate how it bypasses the well-known problems of Taylor expansions. Next, in Section 3 we demonstrate the significant solution capabilities of our procedure. We calculate numerical solutions for several problems previously studied by others. In comparing our results to the prior studies, in general, our solutions are more accurate. This fact, perhaps more than anything else, proves the unique capabilities of the procedure, in spite of its foundation based on Taylor series components. In Section 4 we summarize our conclusions and provide suggestions for future developments.

One final point: we use a commercially available spreadsheet to solve the problems presented in this paper, and that may seem to be unusual. Although not optimized for the task, the spreadsheet is nevertheless effective. It very quickly and conveniently demonstrates the capabilities and features of our procedure.

2. DESCRIPTION OF THE INTEGRAL EQUATION SOLUTION PROCEDURE

Our procedure solves a system of integral equations by constructing a set of mathematically linked Taylor series coefficients for all the functions that are involved in the equation system. Mathematically, it is equivalent to reconstructing each equation system into an isomorphic set of Taylor series coefficient functions (the coefficients automatically update to new values as different expansion points are selected). Reconstruction is possible because an isomorphism exists between any set of analytic functions and their corresponding set of Taylor coefficients, with the coefficients all evaluated at the same point. Stated differently, for each operation that can be performed on a given set of analytic functions, an equivalent operation exists for the corresponding set of Taylor coefficients all evaluated at the same point. Therefore, operations of addition, subtraction, multiplication, division, square root, matrix inverse, partial differentiation, and partial integration, among others, are defined for Taylor coefficients. Using these operations, one can reconstruct any integral equation, involving analytic functions only, strictly in terms of interacting Taylor coefficients. In this paper we will refer to the isomorphic operations involving Taylor coefficients as "Taylor addition," "Taylor multiplication," "Taylor differentiation," etc.

We can easily illustrate the isomorphic operations for Taylor series coefficients by referring to functions of a single variable. For example, let f, g, and h be three functions of x such that

$$f = f(x) = F_i(x - xO)^i \tag{1}$$

$$g = g(x) = G_j(x - xO)^j \tag{2}$$

$$h = h(x) = H_k(x - xO)^k. \tag{3}$$

All three functions are analytic at some point $x = xO$. F_i, G_j, and H_k are the Taylor coefficients of f, g, and h, respectively, evaluated at xO. Above, we adopt the Einstein convention of using a repeated index to indicate a summation over that index. Assume now that f, g, and h are related through multiplication. Specifically, let

$$fg = h. \tag{4}$$

If so, the three sets of Taylor coefficients are related by Taylor multiplication. That is, by

substituting (1), (2), and (3) into (4), performing the multiplication, and then equating the coefficients of the same powers of $(x - xO)$ on both sides of the equation, we must have

$$H_i = \sum_{all\ j,k}^{j+k=i} F_j G_k. \tag{5}$$

Above, the summation is over all j, k such that $j + k = i$. Assuming (1) through (3) are true, (5) is true if and only if (4) is true. In other words, (5) defines Taylor multiplication for functions of a single variable. For functions of four variables Taylor multiplication becomes

$$H_{abcd} = \sum_{all\ i,n}^{i+n=a} \left[\cdots, \left[\cdots, \left[\sum_{all\ m,q}^{m+q=d} F_{ijkm} G_{nopq} \right] \right] \right]. \tag{6}$$

Next, Taylor division can be defined, using multiplication and the multiplicative inverse. Staying with functions of a single variable, let

$$j = j(x) = J_k (x - xO)^k \tag{7}$$

be the multiplicative inverse of f. If so, we then have

$$fj = jf = f/f = f(1/f) = 1. \tag{8}$$

Equation (8) defines the multiplicative inverse function, $j = j(x) = 1/f$. Assuming that (1) and (7) are true, and that f does not vanish at xO, then (8) is true if and only if

$$J_0 = 1/F_0; \qquad J_n = -\left(J_0 F_n + \sum_{all\ k,m>0}^{k+m=n} J_k F_m \right) \Big/ F_0; \ \text{all}\ n > 0. \tag{9}$$

Equation (9) defines Taylor division (Taylor multiplicative inverse) for single variable functions. The equation for J_0 simply states that the first Taylor coefficient for fj equals 1. The equation for J_n, $n > 0$, states that the all the remaining Taylor coefficients in (8) are identically zero. Equation (9) is a recursion formula, because it defines J_n in terms of J_m, for each $n > m$. The spreadsheet handles recursion formulas extremely well. Note that if f vanishes at xO, (8) cannot be analytically satisfied there, and the Taylor coefficients for $j = 1/f$ do not exist at xO.

Next, let $s = s(x)$ equal the square root of f. Therefore:

$$s = s(x) = S_i (x - xO)^i \tag{10}$$

$$s^2 = [s(x)]^2 = f = f(x). \tag{11}$$

We can write a recursion equation for S_i by first substituting F_i for H_i on the left hand side of (5), and S_j and S_k for F_j and G_k, respectively, on the right hand side. Then, solving for S_i and assuming that f does not vanish at xO, we obtain

$$S_0 = \pm\sqrt{F_0}; \qquad S_n = \left[F_n - \sum_{all\ k,m>0}^{k+m=n} S_k S_m \right] \Big/ (2 * S_0); \ n > 0. \tag{12}$$

If $F_0 = 0$, we can still write a formula for the Taylor square root. The procedure is the same as the one identified above, except that we have to use the first non-vanishing Taylor coefficient in the $f = f(x)$ expansion. Note that if $F_0 = 0$, the second coefficient, F_1, must also vanish (otherwise, f does not possess a real valued square root function at xO).

Next, let $d = d(x)$ equal the derivative of f. Therefore

$$d = d(x) = df/dx = D_i(x - xO)^i. \tag{13}$$

Obviously

$$D_i = (i + 1) * F_{i+1}; \qquad i \geq 0. \tag{14}$$

For functions of two or more variables (14) generalizes to a Taylor partial derivative. For example, with four variables and differentiating on the second one,

$$D_{ijkm} = (j + 1) * F_{i(j+1)km}; \qquad j \geq 0. \tag{15}$$

Next, let $i = i(x)$ equal the integral of $f = f(x)$. Therefore

$$i = i(x) = \int_{xO}^{x} f(z)\,dz + C = I_j(x - xO)^j \tag{16}$$

$$I_0 = C; \qquad I_j = F_{j-1}/j; \ j > 0. \tag{17}$$

Equation (17) defines Taylor integration for functions of a single variable. $I_0 = C$ is the constant of integration. In our procedure we refer to it as a primary Taylor coefficient. For functions of four variables with integration on the third one, (17) generalizes to

$$I_{jk0n} = C_{jk0n}; \qquad I_{jkmn} = F_{jk(m-1)n}/m; \ m > 0. \tag{18}$$

Again the I_{jk0n} are primary Taylor coefficients.

One or more primary coefficients occur each time we perform Taylor integration. Suppose we are integrating an Nth order differential system. If so, primary coefficients will occur in the solution functions and their various derivatives of order $(N - 1)$ or less. As shown above, they correspond to Taylor expansion terms that are independent of the integration variable. Primary coefficients are usually determined by initial and/or boundary conditions. For convenience we refer to all other, non-primary coefficients within the solution algorithm as secondary coefficients.

Some of the above equations are not new. All of them, however, are based on our ability to perform term-by-term mathematical operations, such as addition, multiplication, and differentiation, on any Taylor series. The theoretical basis for these operations is well known. Several standard texts, including [1], provide the mathematical justification. Also, the power series solution method, which is well known, uses some of these operations to solve many types of ordinary differential equations (ODEs). In our procedure, however, we take it several steps further. For example, the power series method does not routinely include divisions or square roots. Those operations are usually too complicated, because of the way that method is implemented. As a result of this and other implementation issues, the power series method is usually restricted to fairly simple ODE problems. In contrast,

with our procedure we can solve a much wider range of problems, including systems of nonlinear partial differential equations.

To provide for that capability, our procedure also includes the ability to create Taylor coefficients for several analytical functions, such as natural logarithms, exponents, sines, and cosines. For example, let $l = l(x)$ equal the natural logarithm of f:

$$l = l(x) = \ln(f) = \ln(f(x)) = L_i(x - xO)^i. \tag{19}$$

The L_i are the coefficients for the Taylor expansion, where the expansion point is xO ($f(x) > 0$). Therefore: $L_0 = \ln(f(xO))$. We create the remaining coefficients, L_j ($j > 0$), by constructing coefficients for the function, $(df/dx)/f$. We do that by creating the coefficients for f and df/dx separately, and then performing the indicated Taylor division. That is, instead of dividing df/dx by f, we perform the coefficient space equivalent operation of Taylor dividing the coefficients of df/dx by the coefficients of f. Specifically, using (9), we construct the coefficients for the multiplicative inverse function, $1/f$. Then, we Taylor multiply the coefficients of df/dx and $1/f$, using (5). Note that we create the coefficients of df/dx as the Taylor derivative of the f coefficients. Let C_i be the coefficients for $(df/dx)/f$, constructed as stated above, where $i = 0, 1, 2, \ldots,$. If so, C_i and L_j are related as

$$L_j = ((j-1)!/j!) * C_{(j-1)} = C_{(j-1)}/j; \qquad j = 1, 2, \ldots. \tag{20}$$

The factorial factors adjust for the $1/(j-1)!$ factor contained in $C_{(j-1)}$, versus the $1/j!$ factor required in L_j. If f is a function of several variables, x, y, z, and t, for example, we modify the above procedure to include the coefficients from all of the first order partial derivative functions, $(\partial f/\partial x)/f$, $(\partial f/\partial y)/f$, $(\partial f/\partial z)/f$, and $(\partial f/\partial t)/f$. Then, when we construct the remaining Taylor coefficients for the logarithm (the ones that follow the first one), we must not double, triple, or quadruple the various terms in each coefficient. Since differentiation is independent of the order of the derivatives, that will happen, if we simply add together various coefficient terms from the four first order derivatives. Correctly adding the terms, for functions of four variables, (20) becomes

$$L_{ijkm} = \left\{{}_1C_{(i-1)jkm}/i + {}_2C_{i(j-1)km}/j + {}_3C_{ij(k-1)m}/k + {}_4C_{ijk(m-1)}/m\right\}/N. \tag{21}$$

Above, the ${}_1C_{(i-1)jkm}$ are the Taylor coefficients from $(\partial f/\partial x)/f$, the ${}_2C_{i(j-1)km}$ are the Taylor coefficients from $(\partial f/\partial y)/f$, etc. Also, ${}_1C_{(i-1)jkm} = 0$, if $(i-1) < 0$; ${}_2C_{i(j-1)km} = 0$, if $(j-1) < 0$; etc. Finally, $N = 4$, if none of the four Taylor coefficients, ${}_1C_{(i-1)jkm}$, ${}_2C_{i(j-1)km}$, etc., equals zero. $N = 3$, if exactly one of the four coefficients equals zero, etc. At least one of the four coefficients must always be nonzero, since we are creating logarithm coefficients that follow after the first one, L_{0000} ($= \ln(f(xO, yO, zO, tO))$).

In a similar fashion we can construct coefficients for $e^{f(x)}$, $\sin(f(x))$, and $\cos(f(x))$. To save space we will not provide the details here.

For convenience we refer to each Taylor coefficient as having a level N, where $N = 0, 1, 2, \ldots,$. The level is equal to the sum of the exponents that act on the independent variables, when we look at the one and only term within the expansion from which that coefficient came. For example, the coefficient, F_{ij}, is associated with a function of two independent variables and has a level equal to $i + j$.

At this point we can transfer the above mathematics to a spreadsheet. It is the spreadsheet that makes it all practical for solving differential or integral equations. To see how that works, suppose we have the differential equation,

$$0 = d^2u/dx^2 - (du/dx)/x - 8 * x^2 * u^3, \tag{22}$$

with initial conditions $u(0) = 1$ and $du(0)/dx = u,x(0) = 0$. Even though (22) is an ODE, it is still somewhat challenging. Using it as an illustrative example, we can demonstrate the major features of our procedure. Note that the exact solution is $u = 1/(1 + x^2)$. Also, the Taylor expansion of the solution at $x = 0$ has a radius of convergence of only 1.

To solve any differential system we must first convert it to an equivalent integral equation system. Usually there are several ways that can be done. In Eq. (22), we could solve for $du/dx = u,x$ and integrate once. Or, trivially, we could solve for u and integrate zero times. For reasons that we will discuss later, neither of those approaches will work here. Therefore instead, we solve for the highest order derivative, $d^2u/dx^2 = u,xx$, and then integrate twice. We obtain

$$u = \int_{xO}^{x} \left(\int_{yO}^{y} ((u,z)/z + 8 * z^2 * u^3)\, dz \right) dy + C_1(x - xO) + C_0. \tag{23}$$

C_0 and C_1 are the integration constants. For our specific initial value problem, they equal 1 and 0, respectively (and, $xO = 0$). For the moment we will keep C_0 and C_1 arbitrary so that (23) remains as the general solution to (22). By direct differentiation one can verify that (23) satisfies (22) identically.

Next, we reconstruct (23) on a spreadsheet, substituting the isomorphic Taylor equivalents for the functions and function-based operations shown in the equation. We use (5), (9), (14), and (17) to construct the Taylor equivalents. Figure 1 illustrates how it is done. In the figure the row numbers for the spreadsheet cells are shown at the far left, starting with row 1 at the top and reading down. The column IDs are shown at the top of the figure, starting with the wide column A on the far left and reading from left to right to the less wide columns B, C, D, etc. The value of the expansion point coordinate, xO, is in cell B7. The value of the increment, dx, used for calculating Taylor partial sums, is in cell C7. The Taylor coefficients for the various functions involved in the solution are in rows 11 through 23, starting in column B and proceeding to the right for columns C, D, E, etc. Cells A11 through A23 identify the particular functions for which the Taylor coefficients are calculated. Row 1 identifies the values of the exponent on $(x - xO)$ for each coefficient in the Taylor expansion of each function. For example, $u(x)$ has a Taylor expansion term of $-0.019996 * (x - 0.01)^1$, because cell C11 has the value of -0.019996, cell A11 identifies the associated analytic function as $u(x)$, cell B7 identifies the expansion point as 0.01, and cell C1 identifies the exponent on $(x - xO) = (x - 0.01)$ as 1. Similarly, $u,x = \partial u/\partial x$ has a Taylor series term of $+0.11994 * (x - 0.01)^2$, because cell D12 has the value of $+0.11994$, cell A12 identifies the function as u,x, etc.

Using Taylor operations, we create all of the necessary coefficients to reconstruct (23). By surveying column A, we can match the coefficients of the various stated functions against the actual functions contained in (23), or the functions required to construct it. For example, in row 12 we have Taylor coefficients for u,x, constructed by Taylor differentiation of the u coefficients in row 11. In keeping with this example, let U_i and UX_j be the Taylor coefficients for u and u,x, respectively; $i, j = 0, 1, \ldots$. Selecting cell D12 at random,

	A	B	C	D	E	F
1		0	1	2	3	4
2						
3	*Solution of the ordinary differential equation (ODE): 0 = u,xx - (u,x)/x - 8*x^2*u^3*					
4					-0.0199960	0.99990000
5						
6		*x0*	dx		*ux.0*	*u.0*
7		0.01	0		-0.0199960	0.99990000
8						
9						
10	Taylor solution coefficients, rows 11 through 23					
11	*u = u(x)*	0.99990000	-0.0199960	-0.9994001	0.03998000	0.99850069
12	*u,x = du/dx*	-0.0199960	-1.9988002	0.11994001	3.99400279	-0.2997201
13	*u,xx = d^2u/(dx)^2*	-1.9988002	0.23988003	11.9820083	-1.1988805	-29.916062
14	*x*	0.01	1			
15	*1/x*	100	-10000	1000000	-1.00e+08	1.00e+10
16	*x^2*	0.0001	0.02	1		
17	*u^2*	0.99980002	-0.0399880	-1.9982005	0.11992002	2.99400349
18	*u^3*	0.99970005	-0.0599760	-2.9964014	0.23980008	5.98501049
19	*(u,x)/x*	-1.9996000	0.07997600	3.99640119	-0.2398400	-5.9880069
20	*x^2*u^3*	0.00009997	0.01998800	0.99820089	-0.1198800	-2.9910069
21	*(u,x)/x + 8*x^2*u^3 (= u,xx)*	-1.9988002	0.23988003	11.9820083	-1.1988805	-29.916062
22	*INTEG[u,xx*dx] (=u,x)*	-0.0199960	-1.9988002	0.11994001	3.99400279	-0.2997201
23	*INTEG[INTEG[...,]] (= u)*	0.99990000	-0.0199960	-0.9994001	0.03998000	0.99850069
24						
25		-0.0199960				
26		0.99990000				
27						
28	*Value of x*	0.01	0.02	0.03	0.04	0.05
29	*Exact Solution, u = 1/(1 + x^2)*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
30	*Non Propagated Solution*					
31	*Calc. Solution Error*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
32						
33	*Propagated Solution*					
34	*Calc. Solution Error*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623

FIG. 1. Portion of solution spreadsheet for $0 = u, x, x - (u, x)/x - 8 * x^2 * u^3$ illustrative example.

this cell contains the value of the UX_2 coefficient, because cell D1 has the value of 2. As a result cell D12 must have the formula $= 3 * \text{E11}$. That is, using (14), which is Taylor differentiation, cell D12 must have a formula equivalent to the equation $UX_2 = 3 * U_3$. Cell E11 contains the value of the U_3 coefficient, because cell E1 has the value of 3. Hence, the formula $=3 * \text{E11}$ in cell D12.

In a similar fashion every other row contains the coefficients for the function indicated in column A of that row. Except for row 14, the coefficients in each row are obtained by applying the appropriate Taylor operation to the coefficients in other rows. Specifically, row 13 is the Taylor derivative of row 12. Row 14 has the Taylor coefficients for $f(x) = x$, evaluated at xO. Row 15 is the Taylor multiplicative inverse of row 14 (supplying the coefficients for $j(x) = 1/x$, evaluated at xO). Row 16 is the Taylor multiplication of row 14 with row 14 (coefficients for x^2). Row 17 is the Taylor multiplication of row 11 with row 11 (coefficients for u^2). Row 18 is the Taylor multiplication of rows 11 and 17 (coefficients for u^3). Row 19 is the Taylor multiplication of rows 12 and 15 (coefficients for $(u, x)/x$). Row 20 is the Taylor multiplication of rows 16 and 18 (coefficients for $x^2 * u^3$). Row 21 equals row 19 plus 8 times row 20, (thus obtaining coefficients for $(u, x)/x + 8 * x^2 * u^3$, fully equivalent to solving (22) for u, x, x). Row 22 is the Taylor integral of row 21 (obtaining

the coefficients for u, x through Taylor integration). Row 23 is the Taylor integral of row 22 (the coefficients for u). Finally, we complete the reconstruction of (23) by equating row 11 to row 23. That is, we put the "equal sign" in the equation. The completed spreadsheet has Taylor coefficients to level 8 for the solution function, $u = u(x)$. Eight levels is only a modest number of coefficients.

Note that rows 18 and 20 both contain formulas for Taylor multiplication. Therefore, we could have created the formulas in row 20 by copying from row 18, and, using the spreadsheet's "Find and Replace" utility, change the row numbers to 16 and 18 in all the copied formulas. The resulting formulas, now edited, would be correct for row 20, because this row is the Taylor multiplication of rows 16 and 18. The "Find and Replace" utility is very fast. We can use it to edit any Taylor formula for specific use. Therefore, once we create a formula, we can place it in a spreadsheet library for use in solving other differential or integral problems. It in fact becomes a generic formula for that particular Taylor operation. The only requirement for successful use is that we keep the same arrangement of the Taylor coefficients on all the spreadsheets. Of course, we need a separate set of generic formulas for each different dimension of the independent variable space.

Continuing with the spreadsheet, cells B22 and B23 contain the values of the primary coefficients. The value in cell B22 must equal C_1 from (23), while the value in cell B23 must equal C_0.

By examining the spreadsheet and the formulas on it, one can demonstrate that the Taylor coefficients have the following properties:

(1) The secondary coefficients are functions of only the primary coefficients and the expansion point coordinates. Therefore, once the expansion point and the primary coefficients are specified, all of the Taylor coefficients are then known at that particular point.

(2) When considered as a whole, the coefficients of any level M are all independent of coefficients from any level N, where $N > M$.

(3) Neglecting truncation and rounding errors, all the Taylor coefficients have exact values, for a particular solution at a particular expansion point, if and only if the primary coefficient values are exact for that solution at that point.

Item (2), above, is of particular importance. It means that all of the coefficients are interconnected and form a complete system of recursion formulas. That is critical because we are creating formulas only up to some finite maximum level M. Therefore because of item (2), we do not have the situation where some of our coefficients are dependent on other coefficients from a level N, where $N > M$. If that were to happen, those coefficients would have inaccurate values, because we stopped at level M. For our particular differential problem item (2) would not be true, if we attempted to solve (22) for u, x (or u), and then integrate once (or zero times). For example, if we had solved for u, x, (23) would have become

$$u = \int_{xO}^{x} (z * u, z, z - 8 * z^3 * u^3)\, dz + C_0. \tag{24}$$

The first term under the integral sign causes the problem. From the point of view of Taylor coefficients, (24) states that the level M (the maximum level) coefficient of u is dependent on the level M coefficient of the integral of $z * u, z, z$. Taylor multiplication causes no problem, but Taylor differentiation and integration potentially do. For each integration a Taylor coefficient increases in level by 1. For each differentiation a coefficient decreases in

level by 1. Therefore, the level M coefficient of the integral of $z * u, z, z$ is dependent on the level $(M-1)$ coefficients of x and u, x, x. The level $(M-1)$ coefficient of u, x, x, however, is dependent on the level $(M+1)$ coefficient of u, which we did not construct.

For item (2) to remain true, we require the following integration rule for any system of integral equations:

> *The integrand of any integral cannot directly contain derivatives of the solution functions that are of higher order than the number of multiplies of which that integral is a part.*

If, for example, $v = v(x, y)$ is the solution for a particular problem, the multiple integral,

$$\int \partial v/\partial x \left[\int (\partial^2 v/\partial x\, \partial y)\, dx \right] dy,$$

satisfies the rule. Conversely, the multiple integral,

$$\int \partial^2 v/\partial x\, \partial y \left[\int (\partial v/\partial x)\, dx \right] dy,$$

does not, since the outermost integral is in essence a single integral, yet it contains the second order derivative, $\partial^2 v/\partial x \partial y$, directly within its integrand. Complex arrangements can occur within a given system of integral and/or differential equations. Our integration rule applies to all such situations. We assume that we can always integrate any differential system to satisfy the rule. Conversely, if an integral system does not satisfy it, we assume that we can differentiate, manipulate the terms, and then integrate to satisfy.

Next, Fig. 2 helps to illustrate items (1) and (3). In the figure we randomly set the expansion point and the primary coefficients as $xO = 1$, $u(xO) = 5$, and $u, x(xO) = 3$ (cells B7, B23, and B22, respectively). Note the values of all the other derivatives of u, u, x and u, x, x. One can easily verify the accuracy of all the values by repeatedly differentiating (23) (or, (22)) as many times as required, thus generating a series of recursion formulas, and then substituting in the values of the expansion point and previously obtained derivatives, beginning with the given values of $u(xO)$ and $u, x(xO)$.

Finally, we can demonstrate the ability of our procedure to calculate accurate solutions at considerable distances from the initial condition expansion point. To do that we use a new technique, called "expansion point propagation." Figures 3–5 help illustrate the method. In Fig. 3 we begin at a starting expansion point with starting values for u and u, x. Normally, we use the given initial condition values for the start conditions. For our particular problem, however, the second order equation, (22), contains the function, $1/x$, which is unbounded at the initial expansion point, $xO = 0$. Fortunately, this function multiplies u, x, which vanishes at $xO = 0$. Therefore, we can start fairly close to the initial values without too much trouble. In Fig. 3, we chose the exact solution, $xO = 0.01$, $u(0.01) = 1/(1 + 0.01^2) \approx 0.99990001$ and $u, x(0.01) = -0.02/(1+0.01^2)^2 \approx -0.019996001$, as the start values. Cells B7, B23, and B22 respectively contain these numbers. Because we started with exact values, Fig. 3 shows the exact values of all the Taylor coefficients at the selected start point $xO = 0.01$. That is true because of items (1) and (3), above.

	A	B	C	D	E	F
1		0	1	2	3	4
2						
3	*Solution of the ordinary differential equation (ODE): 0 = u,xx - (u,x)/x - 8*x^2*u^3*					
4					-0.0199960	0.99990000
5						
6		*x0*	dx		*ux.0*	*u.0*
7		1	0		3	5
8						
9						
10	Taylor solution coefficients, rows 11 through 23					
11	*u = u(x)*	5	3	501.5	800	25665
12	*u,x = du/dx*	3	1003	2400	102660	387079
13	*u,xx = d^2u/(dx)^2*	1003	4800	307980	1548316	48533985
14	*x*	1	1			
15	*1/x*	1	-1	1	-1	1
16	*x^2*	1	2	1		
17	*u^2*	25	30	5024	11009	512952.25
18	*u^3*	125	225	37747.5	105162	5782949.25
19	*(u,x)/x*	3	1000	1400	101260	285819
20	*x^2*u^3*	125	475	38322.5	180882	6031020.75
21	*(u,x)/x + 8*x^2*u^3 (= u,xx)*	1003	4800	307980	1548316	48533985
22	*INTEG[u,xx*dx] (=u,x)*	3	1003	2400	102660	387079
23	*INTEG[INTEG[...,]] (= u)*	5	3	501.5	800	25665
24						
25		3				
26		5				
27						
28	*Value of x*	0.01	0.02	0.03	0.04	0.05
29	*Exact Solution, u = 1/(1 + x^2)*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
30	*Non Propagated Solution*					
31	*Calc. Solution Error*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
32						
33	*Propagated Solution*					
34	*Calc. Solution Error*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623

FIG. 2. Spreadsheet for $0 = u, x, x - (u, x)/x - 8 * x^2 * u^3$ with $xO = 1, u(xO) = 5, u, x(xO) = 3$.

Next, note that cells B25 and B26 respectively contain the formulas

$$= \mathrm{B22} + \mathrm{C22} * dx + \mathrm{D22} * dx^2 + \cdots, + \mathrm{I22} * dx^7 \tag{25}$$

$$= \mathrm{B23} + \mathrm{C23} * dx + \mathrm{D23} * dx^2 + \cdots, + \mathrm{J23} * dx^8. \tag{26}$$

That is, they contain formulas for the Taylor partial sum calculation of u, x, and u, evaluated at a point dx away from $xO = 0.01$. As shown in the figure, we chose $dx = 0.01$ (cell C7). Therefore, cells B25 and B26 respectively contain the estimate for u, $x(0.02)$ and $u(0.02)$. Assume for the moment that these are exact values and not approximate. If so, let us replace the values in cells B22 and B23 with the values from cells B25 and B26, respectively, and also change the value of xO to 0.02 in cell B7. Figure 4 shows those replacements. Then, in keeping with our assumption and again using items (1) and (3), Fig. 4 also shows the exact values of all the Taylor coefficients, now evaluated at the new expansion point, $xO = 0.02$. That is, we have propagated the exact solution from the starting expansion point, 0.01, to the new one at 0.02. That is what Figs. 3 and 4 show, assuming in Fig. 3 that cells B25 and B26 contain exact values.

We know, however, that Taylor partials sums are usually approximate and they are in this case. As a result, the coefficient values in Fig. 4 are also approximate and not exact. Look,

	A	B	C	D	E	F
1		0	1	2	3	4
2						
3	*Solution of the ordinary differential equation (ODE): 0 = u,xx - (u,x)/x - 8*x^2*u^3*					
4					-0.0199960	0.99990000
5						
6		*x0*	dx		*ux.0*	*u.0*
7		0.01	0.01		-0.0199960	0.99990000
8						
9						
10	**Taylor solution coefficients, rows 11 through 23**					
11	*u = u(x)*	0.99990000	-0.0199960	-0.9994001	0.03998000	0.99850069
12	*u,x = du/dx*	-0.0199960	-1.9988002	0.11994001	3.99400279	-0.2997201
13	*u,xx = d^2u/(dx)^2*	-1.9988002	0.23988003	11.9820083	-1.1988805	-29.916062
14	*x*	0.01	1			
15	*1/x*	100	-10000	1000000	-1.00e+08	1.00e+10
16	*x^2*	0.0001	0.02	1		
17	*u^2*	0.99980002	-0.0399880	-1.9982005	0.11992002	2.99400349
18	*u^3*	0.99970005	-0.0599760	-2.9964014	0.23980008	5.98501049
19	*(u,x)/x*	-1.9996000	0.07997600	3.99640119	-0.2398400	-5.9880069
20	*x^2*u^3*	0.00009997	0.01998800	0.99820089	-0.1198800	-2.9910069
21	*(u,x)/x + 8*x^2*u^3 (= u,xx)*	-1.9988002	0.23988003	11.9820083	-1.1988805	-29.916062
22	*INTEG[u,xx*dx] (=u,x)*	-0.0199960	-1.9988002	0.11994001	3.99400279	-0.2997201
23	*INTEG[INTEG[...,]] (= u)*	0.99990000	-0.0199960	-0.9994001	0.03998000	0.99850069
24						
25		-0.0399680				
26		0.99960015				
27						
28	*Value of x*	0.01	0.02	0.03	0.04	0.05
29	*Exact Solution, u = 1/(1 + x^2)*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
30	*Non Propagated Solution*					
31	*Calc. Solution Error*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
32						
33	*Propagated Solution*	0.99990000				
34	*Calc. Solution Error*	1.1102E-16	0.99960015	0.99910080	0.99840255	0.99750623

FIG. 3. Propagation spreadsheet at start expansion point, $xO = 0.01$.

however, at cell C34 in Fig. 4. This cell contains the difference between the exact solution at $xO = 0.02$ (cell C29) and the approximate one, obtained from the Taylor partial sum from cell B26 in Fig. 3. The approximate solution is in fact very accurate; the error is only of order 10^{-14}. Therefore, we can repeat the propagation procedure several times without accumulating significant error. In fact, we can repeat it at least 253 more times, propagating the expansion point from $xO = 0.02$ to $xO = 2.55$. Figure 5 shows those results. In the figure the first five columns, starting from the left, show the calculations for values of xO between 0.01 and 0.04 (rows 28, 29, 33, and 34 on the spreadsheet). Conversely, the last column on the right, column IV, gives the results at $xO = 2.55$. The maximum error, shown in row 34, occurs at $xO = 2.55$ and is of order 10^{-11}. Therefore, we have generated accurate solution values far beyond the initial radius of convergence of 1. We used a macro, a spreadsheet user written program, to automate the task of replacing values in cells B22, B23, and B7.

Figure 5 also shows the results obtained by the usual and well-known method of Taylor partial sum expansions, where the expansion point remained fixed at $xO = 0.01$ (no propagation). The calculated values and corresponding errors are in rows 30 and 31, respectively. The errors are very large at and near $xO = 2.55$. These large errors occur because the Taylor

	A	B	C	D	E	F
1		0	1	2	3	4
2						
3	*Solution of the ordinary differential equation (ODE): 0 = u,xx - (u,x)/x - 8*x^2*u^3*					
4					-0.0199960	0.99990000
5						
6		*x0*	dx		*ux.0*	*u.0*
7		0.02	0.01		-0.0399680	0.99960015
8						
9						
10	Taylor solution coefficients, rows 11 through 23					
11	*u = u(x)*	0.99960015	-0.0399680	-0.9976023	0.07984017	0.99401118
12	*u,x = du/dx*	-0.0399680	-1.9952047	0.23952053	3.97604474	-0.5977640
13	*u,xx = d^2u/(dx)^2*	-1.9952047	0.47904107	11.9281342	-2.3910561	-29.665006
14	*x*	0.02	1			
15	*1/x*	50	-2500	125000	-6250000	312500000
16	*x^2*	0.0004	0.04	1		
17	*u^2*	0.99920047	-0.0799040	-1.9928095	0.23936089	2.97605591
18	*u^3*	0.99880095	-0.1198081	-2.9856239	0.47840268	5.94016771
19	*(u,x)/x*	-1.9984009	0.15980815	3.98561918	-0.4787217	-5.9521118
20	*x^2*u^3*	0.00039952	0.03990411	0.99281438	-0.2390417	-2.9641117
21	*(u,x)/x + 8*x^2*u^3 (= u,xx)*	-1.9952047	0.47904107	11.9281342	-2.3910561	-29.665006
22	*INTEG[u,xx*dx] (=u,x)*	-0.0399680	-1.9952047	0.23952053	3.97604474	-0.5977640
23	*INTEG[INTEG[...,]] (= u)*	0.99960015	-0.0399680	-0.9976023	0.07984017	0.99401118
24						
25		-0.0598921				
26		0.99910080				
27						
28	*Value of x*	0.01	0.02	0.03	0.04	0.05
29	*Exact Solution, u = 1/(1 + x^2)*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
30	*Non Propagated Solution*					
31	*Calc. Solution Error*	0.99990000	0.99960015	0.99910080	0.99840255	0.99750623
32						
33	*Propagated Solution*	0.99990000	0.99960015			
34	*Calc. Solution Error*	1.1102E-16	2.5535E-14	0.99910080	0.99840255	0.99750623

FIG. 4. Propagation spreadsheet at next expansion point, $xO = 0.02$.

partial sum in (26) contains terms only up to level 8. For accurate results many more terms are needed as the distance from the expansion point increases. For this problem note that the radius of convergence equals 1 only if the expansion point is $xO = 0$; at all other points the radius is infinite. In summary, the propagation method produced substantial increases in solution accuracy over the desired range of x.

Based on limited research, however, propagation does not always work exactly as shown above. For some problems it requires adaptive step sizes (dx in the above illustration) for portions of the solution where the rates of change are large. For other problems we were not able to use the technique to calculate accurate solutions over the entire desired range of the variables. In several of those cases, however, we successfully used the non-propagation method. In order for that to work the radius of convergence has to be sufficiently large at the fixed expansion point. For problems involving a single independent variable we can usually estimate the convergence radius, using the Cauchy–Hadamard formula [1]. We simply construct a solution spreadsheet with a sufficient number of Taylor coefficients to use the formula for an accurate estimate. For multivariable problems we would require a similar formula. Additional research is therefore needed to derive the formula and to place expansion point propagation on a solid theoretical basis.

	A	B	C	D	E	IV
1		0	1	2	3	
2						
3	*Solution of the ordinary differential equation (ODE): 0 = u,xx - (u,x)/x - 8*x^2*u^3*					
4					-0.0199960	
5						
6		*x0*	dx		*ux.0*	
7		2.55	0.01		-0.0906062	
8						
9						
10	Taylor solution coefficients, rows 11 through 23					
11	*u = u(x)*	0.13328890	-0.0906062	0.04382578	-0.0177147	
12	*u,x = du/dx*	-0.0906062	0.08765157	-0.0531443	0.02480222	
13	*u,xx = d^2u/(dx)^2*	0.08765157	-0.1062887	0.07440668	-0.0370757	
14	*x*	2.55	1			
15	*1/x*	0.39215686	-0.1537870	0.06030862	-0.0236504	
16	*x^2*	6.5025	5.1	1		
17	*u^2*	0.01776593	-0.0241536	0.01989247	-0.0126641	
18	*u^3*	0.00236800	-0.0048291	0.00561852	-0.0048636	
19	*(u,x)/x*	-0.0355318	0.04830723	-0.0397849	0.02532830	
20	*x^2*u^3*	0.01539793	-0.0193245	0.01427395	-0.0078005	
21	*(u,x)/x + 8*x^2*u^3 (= u,xx)*	0.08765157	-0.1062887	0.07440668	-0.0370757	
22	*INTEG[u,xx*dx] (=u,x)*	-0.0906062	0.08765157	-0.0531443	0.02480222	
23	*INTEG[INTEG[...,]] (= u)*	0.13328890	-0.0906062	0.04382578	-0.0177147	
24						
25		-0.0897350				
26		0.13238720				
27						
28	*Value of x*	0.01	0.02	0.03	0.04	2.55
29	*Exact Solution, u = 1/(1 + x^2)*	0.99990000	0.99960015	0.99910080	0.99840255	0.13328890
30	*Non Propagated Solution*	0.99990000	0.99960015	0.99910080	0.99840255	1541.77520
31	*Calc. Solution Error*	1.1102E-16	2.5535E-14	-4.4408E-14	4.3076E-14	-1541.6419
32						
33	*Propagated Solution*	0.99990000	0.99960015	0.99910080	0.99840255	0.13328890
34	*Calc. Solution Error*	1.1102E-16	2.5535E-14	5.5511E-14	4.3076E-14	0.00000000

FIG. 5. Propagation spreadsheet at last expansion point, $xO = 2.55$.

3. SIGNIFICANT SOLUTION EXAMPLES

One can easily build the solution spreadsheet for the example problem of Section 2. That is true even if you had to initially construct the various formulas (no generic formulas initially available). Without a doubt, however, there are many other problems that are more significant and much more challenging. Our procedure solves many of them extremely well. As an illustration, shown below are solutions for the following problems: (1) the first problem of the biharmonic equation; (2) the nonlinear Klein–Gordon equation; and (3) shape-from-shading viscosity problems.

3.1. *The First Biharmonic Problem*

In rectangular coordinates the first problem of the biharmonic equation is

$$\nabla^2\nabla^2 u = (\partial^2/\partial x^2 + \partial^2/\partial y^2)^2 \cdot u = f(x, y), \tag{27}$$

where u and its normal derivative, $\partial u/\partial n$, are prescribed on the boundaries of a rectangular region. Several researchers have examined this problem and developed solution algorithms, including Marinos [2] and earlier Stephenson [3]. Building on this work, we tested the

viability of our procedure by solving the following four specific problems:

$$\nabla^2\nabla^2 u = 8 \tag{28}$$

$$\nabla^2\nabla^2 u = 0 \tag{29}$$

$$\nabla^2\nabla^2 u = 8[3x^2(1-x)^2 + 3y^2(1-y)^2 + (6x^2 - 6x + 1)(6y^2 - 6y + 1)] \tag{30}$$

$$\nabla^2\nabla^2 u = (2\pi)^4[4\cos(2\pi x)\cos(2\pi y) - \cos(2\pi x) - \cos(2\pi y)]. \tag{31}$$

For the above, the exact solutions are $u = x(1-x)y(1-y)$; $u = x^2 - y^2 + x\mathrm{e}^x \cos y$; $u = x^2(1-x)^2y^2(1-y)^2$; and $u = [1 - \cos(2\pi x)][1 - \cos(2\pi y)]$, respectively.

For each problem we calculated a numerical solution at several points within the closed rectangular area, $0 \le x, y \le 1$, and then compared the results to the exact solution. We constructed one solution algorithm to solve all four problems, one problem at a time. It has equations for Taylor coefficients to level 16. For the first and third problems we only needed coefficients to level 8. The coefficients between levels 9 and 16 inclusive were not used for calculating their solutions. For the second and fourth problems we needed greater precision. We therefore used all the coefficients to solve these two problems.

To solve (27) through (31) we started with exact values for the primary coefficients at the four corners of the rectangular region. We then used a Taylor partial sum expansion to calculate solutions for various points in the interior, dividing the region into quarters and using the corner of each quarter as a fixed expansion point. Note that this is the usual, well-known use of Taylor series expansions. It does not involve expansion point propagation. In fact, for the first biharmonic problem, propagation produces less accuracy. Therefore, we did not use it here.

Tables 1 and 2 show the results of our simulations. Table 1 shows the difference between the calculated and exact solutions for the first and second problems at intervals of 0.1 in the x and y directions (121 calculation points for each problem). Table 2 shows the difference between calculated and exact for the third and fourth problems, again at intervals of 0.1.

For the first test problem the maximum error in the calculated solution is approximately 1.4×10^{-17}. Solution accuracy therefore is at least 10 orders of magnitude better than results reported by other researchers. The reason for this is that the exact solution, $u = x(1-x)y(1-y)$, is a polynomial of degree 4. Therefore, a Taylor expansion to level 4 can solve this problem exactly, except for truncation and rounding errors. Next, for the second problem the maximum error is 1.6×10^{-10}. This accuracy is 4 to 5 orders of magnitude better than the previous studies. For the third problem maximum error is about 1.9×10^{-10}, which is 4 to 6 orders of magnitude better. Again, the reason for the high degree of accuracy is that the Taylor method solves the problem exactly, except for truncation and rounding, because the exact solution is a polynomial of degree 8. For the fourth problem the maximum error is 7.3×10^{-1}. That result is only one order of magnitude better than the worst case maximum reported in the other studies.

Note that the second and fourth test problems have trigonometric dependencies, which produce oscillations in the primary coefficient boundary condition values. For a given partial expansion these oscillations reduce the accuracy of the resulting calculations. Oscillations are even more important when the expansion point is propagated. That is true because under propagation, partial sum expansions from the first, second, and third order partial derivatives of u enter into the calculations (that is, from $\partial u/\partial x$, $\partial^2 u/\partial x^2$, $\partial^3 u/\partial x^2 \partial y$). These

TABLE 1

$y\backslash x$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$u = x * (1-x) * y * (1-y)$ calculated minus exact solutions no propagation, first biharmonic problem											
0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0	0	3.5E-18	0	0	0	0	3.5E-18	1.7E-18	0
0.2	0	0	0	6.9E-18	0	0	0	0	3.5E-18	3.5E-18	0
0.3	0	3.5E-18	6.9E-18	6.9E-18	6.9E-18	6.9E-18	6.9E-18	6.9E-18	1.4E-17	6.9E-18	0
0.4	0	0	0	6.9E-18	0	0	0	0	6.9E-18	3.5E-18	0
0.5	0	0	0	6.9E-18	0	0	0	0	6.9E-18	6.9E-18	0
0.6	0	0	0	6.9E-18	0	0	0	0	6.9E-18	3.5E-18	0
0.7	0	0	0	6.9E-18	0	0	0	0	6.9E-18	6.9E-18	0
0.8	0	3.5E-18	3.5E-18	1.4E-17	6.9E-18	6.9E-18	6.9E-18	6.9E-18	1E-17	6.9E-18	0
0.9	0	1.7E-18	3.5E-18	6.9E-18	3.5E-18	6.9E-18	3.5E-18	6.9E-18	6.9E-18	5.2E-18	0
1	0	0	0	0	0	0	0	0	0	0	0
$u = x\hat{}2 - y\hat{}2 + x * \mathrm{EXP}(x) * (\cos y)$ calculated minus exact solutions no propagation, first biharmonic problem											
0	0	0	0	1.1E-16	0	0	0	0	−4.4E-16	0	0
0.1	0	−4.5E-15	−7.2E-15	1.6E-13	7.7E-13	2.1E-12	1.6E-10	9.4E-11	4.9E-11	6.7E-12	0
0.2	0	−2.5E-16	−1.2E-14	−4.1E-14	−8.2E-14	−7.9E-14	−6.9E-12	−7.9E-12	−3.7E-12	−1.8E-13	−4.4E-16
0.3	−2.8E-17	−1.7E-16	−1.7E-16	−6.1E-15	−3.3E-14	−9.4E-14	−4.2E-12	−1.8E-12	−4.3E-13	7.4E-14	−4.4E-16
0.4	0	−3E-16	−9.7E-17	1.7E-16	−3.9E-15	−1.9E-14	−6.8E-13	−9.5E-14	−9.3E-15	−1.3E-14	0
0.5	0	−1.1E-16	−2.7E-16	−1.9E-16	−3.9E-16	−1.7E-15	−2.9E-14	1.7E-14	−2.6E-14	−2.7E-14	0
0.6	0	9.4E-16	4.4E-16	1.4E-15	1.8E-14	7E-14	3.8E-13	6.8E-14	1.2E-14	8E-15	0
0.7	0	−6.7E-16	7.4E-15	4.6E-14	1.6E-13	3.8E-13	2.4E-12	9.9E-13	2.4E-13	−2.8E-14	0
0.8	1.1E-16	5.3E-15	1.5E-13	4E-13	5.5E-13	3.8E-13	5.7E-12	4.8E-12	2E-12	1.1E-13	0
0.9	1.1E-16	−1.8E-13	−1.7E-12	−4.4E-12	−1.1E-11	−2.2E-11	−7.6E-11	−4.5E-11	−2.4E-11	−3.3E-12	0
1	0	0	0	1.1E-16	0	5.6E-17	4.9E-17	0	−2.2E-16	−2.2E-16	0

TABLE 2

$y\backslash x$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$u = x\hat{}2 * (1-x)\hat{}2 * y\hat{}2 * (1-y)\hat{}2$ calculated minus exact solutions no propagation, first biharmonic problem											
0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	−7.4E-13	−1.5E-12	2E-11	8.6E-11	1.9E-10	3.1E-12	4.9E-11	8.1E-11	3.4E-11	0
0.2	0	−4.6E-14	−1.5E-12	−2.8E-12	−1.5E-12	3.4E-12	6.3E-13	1E-12	1.1E-12	3.1E-13	0
0.3	0	2E-14	−8.9E-14	−2.9E-13	−4.1E-13	−3.5E-13	3E-15	4.5E-15	4.1E-15	7.5E-16	0
0.4	0	2.6E-15	−1.4E-15	−1.3E-14	−2.3E-14	−2.6E-14	8.2E-18	1.1E-17	1E-17	4.3E-19	0
0.5	0	1.8E-16	1E-16	−3.4E-16	−8.1E-16	−1.1E-15	−4.3E-19	0	6.5E-19	3.3E-19	0
0.6	0	−5.4E-20	0	4.3E-19	4.3E-19	−4.3E-19	3.9E-18	3.5E-18	−4.3E-19	−5.3E-18	0
0.7	0	−1.7E-17	1.3E-17	3.5E-17	3.3E-17	2.2E-17	2.1E-15	2.2E-15	8.2E-16	−1.1E-15	0
0.8	0	−2.4E-15	3.3E-14	5.2E-14	4.4E-14	2.8E-14	7.1E-13	8.4E-13	5.3E-13	−3.8E-14	0
0.9	0	4.3E-12	2.3E-11	3E-11	2.4E-11	1.5E-11	9.6E-11	1.2E-10	9E-11	1.7E-11	0
1	0	0	0	0	0	0	0	0	0	0	0
$u = (1 - \cos 2 * Pl * x) * (1 - \cos 2 * Pl * y)$ calculated minus exact solutions no propagation, first biharmonic problem											
0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	−2.7E-09	5E-08	2.2E-06	4.5E-05	0.000534	4.6E-05	2.2E-06	4E-08	1.4E-10	0
0.2	0	3.9E-08	2.2E-06	5E-05	0.000629	0.00527	0.000629	5E-05	2.2E-06	4E-08	0
0.3	0	2.2E-06	5E-05	0.000634	0.005369	0.033935	0.005369	0.000634	5E-05	2.2E-06	0
0.4	0	4.6E-05	0.000629	0.005369	0.034029	0.172381	0.034029	0.005369	0.000629	4.6E-05	0
0.5	0	0.000536	0.005273	0.033937	0.172381	0.732444	0.172381	0.033937	0.005273	0.000536	0
0.6	0	4.6E-05	0.000629	0.005369	0.034029	0.172381	0.034029	0.005369	0.000629	4.6E-05	0
0.7	0	2.2E-06	5E-05	0.000634	0.005369	0.033937	0.005369	0.000634	5E-05	2.2E-06	0
0.8	0	4E-08	2.2E-06	5E-05	0.000629	0.005273	0.000629	5E-05	2.2E-06	4E-08	0
0.9	0	1.4E-10	4E-08	2.2E-06	4.6E-05	0.000536	4.6E-05	2.2E-06	4E-08	1.4E-10	0
1	0	0	0	0	0	0	0	0	0	0	0

expansions are increasing less accurate with each additional derivative because: (1) the expansions have fewer terms with each added derivative; and (2) the oscillations become larger in magnitude with each derivative. For these reasons the non-propagated calculations are more accurate than the propagated ones for the second and fourth problems.

Because our procedure does not require a solution grid, calculations at one (x, y) point do not depend on those from any other. As a result, our algorithm does not have any of the storage problems reported in the other studies. Also, the algorithm does not need or use iteration to solve the biharmonic equation. As a result, there are no issues or problems concerning solution convergence. Finally, note that we can increase solution accuracy by including more Taylor coefficients on the spreadsheet. In simulations on other problems we constructed and used coefficients to level 32 without any trouble.

3.2. *The Nonlinear Klein–Gordon Equation*

In 2 dimensions the nonlinear Klein–Gordon equation has the form

$$\partial^2 v(x,t)/\partial t^2 - \gamma^2 \partial^2 v(x,t)/\partial x^2 + b\, v(x,t) + g(v(x,t)) = f(x,t) = f \tag{32}$$

$$v(x,0) = v_0(x),\ \partial v(x,0)/\partial t = v_1(x), \tag{33}$$

where $\gamma^2 \geq 0$ and b are real valued constants, $g = g(v(x,t))$ is a given nonlinear real valued function of $v = v(x,t)$ and f is a known real valued function of x and t. We tested our procedure by solving two initial value problems defined by (32) and (33), above, and (34) and (35), below:

$$\gamma^2 = 1, \quad b = -2, \quad g = 0, \quad \text{and} \quad f = -2(\sin x)(\sin t)$$
$$v(x,0) = 0 \quad \text{and} \quad \partial v(x,0)/\partial t = \sin x \tag{34}$$

$$b = c^2; \quad g = -\varepsilon^2 \sigma v^3 \text{ and } f = 0; \quad \gamma, c, \sigma, \text{ and } 0 < \varepsilon \ll 1 \text{ are constants}$$
$$v(x,0) = \cos(kx) \quad \text{and} \quad \partial v(x,0)/\partial t = 0; \quad k \text{ is a constant} \tag{35}$$

When combined with (32) and (33), the solutions for (34) and (35) are, respectively,

$$v = v(x,t) = (\sin x)(\sin t) \tag{36}$$

$$\begin{aligned} v = v(x,t) \cong\ & (\cos \omega t)(\cos kx) + \varepsilon^2[(9\sigma/32\omega)t \sin \omega t \\ & + (3\sigma/128\omega^2)(\cos \omega t - \cos 3\omega t)] \cos kx \\ & + \varepsilon^2[(3\sigma/128\gamma^2 k^2)(\cos \omega t - \cos \lambda t) \\ & + (\sigma/128c^2)(\cos \lambda t - \cos 3\omega t)] \cos 3kx + O(\varepsilon^3);\ \lambda^2 = 9\gamma^2 k^2 + c^2. \end{aligned} \tag{37}$$

Equation (36) is an exact solution, while (37) is an approximate one, derived by perturbation methods.

We built a single algorithm to solve both problems, one at a time. It has coefficients from levels 0 to 8. It solved both problems with excellent precision. Table 3 shows the simulation results. The table has a top and bottom section, each containing 8 rows of data. The top section has the calculation results for the first problem, while the bottom one has the results

TABLE 3
Solution of the Nonlinear Klein–Gordon Equation

	Calculations created by propagating the t coordinate						
	$t\backslash x$	0.1	0.2	0.3	0.4	0.5	0.6
$v = (\sin x) * (\sin t)$: Differences between exact and Taylor prop. calc. sols; $dt = 0.006$	0.314159	8.24E-12	1.64E-11	2.44E-11	3.21E-11	3.96E-11	4.66E-11
	0.628319	4.29E-09	8.54E-09	1.27E-08	1.67E-08	2.06E-08	2.43E-08
	0.942478	1.7E-07	3.38E-07	5.03E-07	6.63E-07	8.16E-07	9.62E-07
	1.256637	2.36E-06	4.7E-06	6.99E-06	9.21E-06	1.13E-05	1.34E-05
	1.570796	1.86E-05	3.7E-05	5.5E-05	7.25E-05	8.92E-05	0.000105
	Calculations created without propagating the t coordinate						
	$t\backslash x$	0.1	0.2	0.3	0.4	0.5	0.6
$v = (\sin x) * (\sin t)$ exact solution	1.570796	0.099833	0.198669	0.29552	0.389418	0.479426	0.564642
Non-propagated solution	1.570796	0.099818	0.198638	0.295474	0.389357	0.47935	0.564554
calculation Solution Error: exact-Taylor non-prop.		1.57E-05	3.12E-05	4.64E-05	6.11E-05	7.52E-05	8.86E-05
	Calculations created by propagating the t coordinate						
	$t\backslash x$	0.1	0.2	0.3	0.4	0.5	0.6
$v = (\cos w * t) * (\cos k * x)^{+} \ldots$,: Differences between pert.-derived and Taylor prop. calc. sols; $dt = 0.005$	0.1	−1.2E-09	−1.1E-09	−9.8E-10	−8.2E-10	−6.4E-10	−4.7E-10
	0.2	−1.9E-08	−1.7E-08	−1.5E-08	−1.3E-08	−1E-08	−7.4E-09
	0.3	−8.8E-08	−8.1E-08	−7.2E-08	−6E-08	−4.8E-08	−3.6E-08
	0.4	−2.5E-07	−2.3E-07	−2.1E-07	−1.8E-07	−1.4E-07	−1.1E-07
	0.5	−5.4E-07	−5.1E-07	−4.5E-07	−3.9E-07	−3.1E-07	−2.4E-07
	Calculations created without propagating the t coordinate						
	$t\backslash x$	0.1	0.2	0.3	0.4	0.5	0.6
$v = (\cos w * t) * (\cos k * x)^{+} \ldots$,	0.5	0.75745	0.746053	0.727187	0.701045	0.667892	0.628064
pert. calc. Non-propagated	0.5	0.75745	0.746053	0.727188	0.701045	0.667892	0.628064
solution calculation. Diff. between pert. calc.-Taylor non-prop.		−4.6E-07	−4.3E-07	−3.9E-07	−3.3E-07	−2.7E-07	−2.1E-07

Note. $c = \gamma = \sigma = +1; \epsilon = +0.01$.

for the second problem. In both sections the first 5 rows show calculation errors for the propagated solution at five equally spaced time intervals and selected values of x. The last 3 rows compare the non-propagated and exact solutions at one value of time and selected x. Looking at all the data, we can easily summarize:

(1) All of the calculations, whether by propagation or not, have small errors.

(2) For system (32), (33), and (34) the calculation errors range from 10^{-12}, at $t = 0.314159$, to 10^{-4} for $t = \pi/2$ ($\cong 1.570796$). The propagated and non-propagated calculations have comparable errors, with the non-propagated calculations slightly better (comparing rows 5 and 8 in the top section of Table 3).

(3) For system (32), (33), and (35), Table 3 shows the differences between the perturbation and Taylor calculations. Differences range from 10^{-10} to 10^{-7}. The propagated and non-propagated Taylor calculations are almost identical (rows 5 and 8 in the bottom section of Table 3).

(4) In comparing to earlier research by Deeba and Khuri [4], our results are more accurate.

3.3. *Shape-from-Shading Viscosity Problems*

For our last set of simulations we examined shape-from-shading viscosity problems. Therefore, let Ω be an area defined on the x, y plane together with its boundary, $\partial\Omega$. The

applicable 2D partial derivative equations are from Hamilton–Jacobi theory:

$$I[1 + (\partial u/\partial x)^2 + (\partial u/\partial y)^2]^{0.5} - \alpha(\partial u/\partial x) - \beta(\partial u/\partial y) - \gamma = 0 \tag{38}$$

$$u = 0 \qquad \text{everywhere on } \partial\Omega. \tag{39}$$

α, β, γ are non-negative constants such that $\alpha^2 + \beta^2 + \gamma^2 = 1$ and $I = I(x, y)$. Under certain assumptions for I and u the above equations have unique solutions for $u = u(x, y)$. Each one defines a surface under illumination from a distant light source. $I = I(x, y)$ is the brightness value on the surface resulting from the illumination. Equations (38) and (39) therefore represent equations for recovering a surface shape, based on knowing the brightness on its surface. Please see Rouy and Tourin [5] for the mathematical theory associated with this equation and its application to shape-from-shading problems.

Consider now the following functions for $I = I(x, y)$, defined on the unit square, $0 \le x$, $y \le 1$, and u defined at points of maximum illumination brightness:

$$\begin{aligned} &I = [1 + (16y(1-y)(1-2x))^2 + (16x(1-x)(1-2y))^2]^{-0.5} \\ &u(0.5, 0.5) = 1; \qquad \text{vertical lighting: } \alpha = \beta = 0 \text{ and } \gamma = 1 \end{aligned} \tag{40}$$

$$\begin{aligned} &I = [1 + (2\pi \sin(2\pi y)\cos(2\pi x))^2 + (2\pi \sin(2\pi x)\cos(2\pi y))^2]^{-0.5} \\ &u(0.25, 02.5) = u(0.75, 0.75) = 1 \\ &u(0.25, 0.75) = u(0.75, 0.25) = -1 \\ &u(0.5, 0.5) = 0; \qquad \text{vertical lighting: } \alpha = \beta = 0 \text{ and } \gamma = 1. \end{aligned} \tag{41}$$

The exact solutions for the above are respectively

$$u = u(x, y) = +16(x^2 - x)(y^2 - y) \tag{42}$$

$$u = u(x, y) = +\sin(2\pi x)\sin(2\pi y). \tag{43}$$

Equation (42) represents a parabolic surface, while (43) is a sinusoidal surface.

We built an algorithm with coefficients from levels 0 to 16 to solve (38) through (41). In the algorithm we used Taylor square roots to convert the Taylor equivalent of (38) into a quadratic formula solution for the coefficients of $u, x = \partial u/\partial x$. We then Taylor integrated the u, x coefficients to obtain coefficients for u. Finally, we calculated solution values at selected x, y points, using partial sum expansions without propagation (the non-propagated calculations were more accurate). We selected various points on $\partial\Omega$ for the expansion points.

Because we solved for the coefficients of u, x directly and then Taylor integrated, our algorithm appears to be more efficient than the approach used in [5]. In particular, we did not have to impose a solution at various points, such as on $\partial\Omega$, or at points of maximum illumination. Therefore, we did not have the solution discontinuities or jumps that were noted in [5]. Our algorithm only required that we determine the sign on the first nonzero square root coefficient at each expansion point. We did that by selecting the sign at the first point, (0, 0), and then calculating its value, $+1$, 0, or -1, at the other expansion points, using Taylor partial sums. We selected at (0, 0), using our knowledge of the exact solution.

TABLE 4

	y values						
x/*y*	0	0.1	0.2	0.3	0.4	0.5	0.6
	Calculation errors for parabola reconstruction: vertical lighting						
0	0	0	0	0	0	0	0
0.1	0	0	2.8E-17	0	0	−5.6E-17	0
0.2	0	−5.4E-15	5.6E-17	0	−1.1E-16	0	−1.1E-16
0.3	0	−2.8E-12	2.3E-15	−1.1E-16	−1.1E-16	−1.1E-16	−1.1E-16
0.4	0	−2.6E-10	1.5E-13	2.2E-16	−1.1E-16	0	−1.1E-16
0.5	0	−8.8E-09	4.4E-12	4.8E-15	2.2E-16	0	2.2E-16
0.6	0	−2.6E-10	1.5E-13	2.2E-16	−1.1E-16	0	−1.1E-16
0.7	0	−2.8E-12	2.3E-15	−1.1E-16	−1.1E-16	−1.1E-16	−1.1E-16
0.8	0	−5.4E-15	−5.6E-17	−1.1E-16	−2.2E-16	−1.1E-16	−2.2E-16
0.9	0	−8.3E-17	−1.4E-16	−2.2E-16	−2.8E-16	−2.8E-16	−2.8E-16
1	0	0	0	0	0	0	0
	Calculation errors for sinusoid reconstruction: vertical lighting						
0	0	0	0	0	0	0	0
0.1	0	4.4E-11	7.2E-11	7.2E-11	4.4E-11	−1.3E-13	−4.4E-11
0.2	0	9E-08	1.5E-07	1.5E-07	9E-08	2.4E-11	−9E-08
0.3	0	7.7E-06	1.2E-05	1.2E-05	7.7E-06	1E-08	−7.7E-06
0.4	0	0.000179	0.000289	0.000289	0.000179	7.8E-07	−0.00018
0.5	0	−0.00204	−0.00329	−0.00329	−0.00204	−2.2E-05	−0.002036
0.6	0	−0.00018	−0.00029	−0.00029	−0.00018	−7.8E-07	0.000179
0.7	0	−7.7E-06	−1.2E-05	−1.2E-05	−7.7E-06	−1E-08	7.7E-06
0.8	0	−9E-08	−1.5E-07	−1.5E-07	−9E-08	−2.4E-11	9E-08
0.9	0	−4.4E-11	−7.2E-11	−7.2E-11	−4.4E-11	1.3E-13	4.4E-11
1	0	0	0	0	0	0	0

Table 4 and Figs. 6 and 7 show the results of our simulations. In the tables we list solution errors at selected values of x and y for (40) and (41). As can be seen the errors are extremely small for the first problem, (40). They range from 10^{-9} to 10^{-17} in magnitude. For the second problem we again see the effect of initial value oscillations on the accuracy. The calculation errors, although still acceptable, are quite a bit larger. They range from 10^{-3} to 10^{-13} in magnitude. Figures 6 and 7 present the shapes reconstructed from (40) and (41). For both simulations the maximum errors are much smaller than those reported in [5].

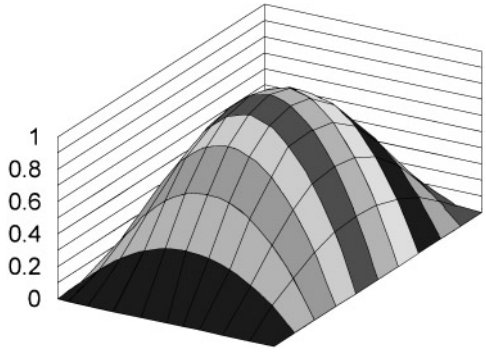

FIG. 6. Parabola reconstruction under vertical lighting.

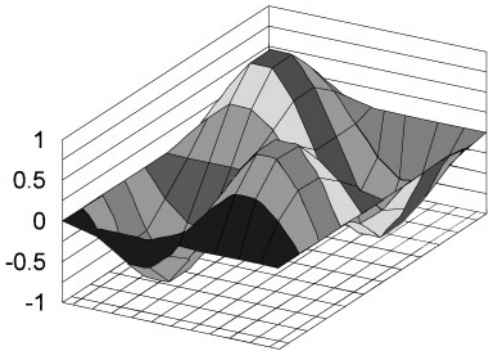

FIG. 7. Sinusoid reconstruction under vertical lighting.

4. SUMMARY AND CONCLUSIONS

In this paper we demonstrated the ability of a new procedure to solve a variety of mathematical and physical problems of interest. Such problems included, but were not limited to, the biharmonic equation and shape-from-shading computations. In all cases, for analytical points only, calculation accuracies were comparable, and most often substantially superior, to accuracies reported in other research. We also demonstrated other key features of the procedure, such as: (1) the ability to solve several related initial value and/or boundary condition problems, using one algorithm; (2) the fact that the independent variable space does not require discretization, which has positive repercussions on storage requirements; (3) the use of generic formulas to build and create solution algorithms that are specific to the problems of interest; and (4) solution accuracies that often are improved by propagating the Taylor expansion points. Additionally, the procedure works, using any given set of independent coordinates for any problem.

In essence the procedure is a machine for numerically generating analytical solutions to integral and differential problems. It does so in Taylor coefficient space. Input an arbitrary set of Taylor coefficients. Dial in the specific mathematical (Taylor) operations to be performed on the coefficients. The machine outputs the resulting solution coefficients that satisfy those operations.

Our procedure does have a few problem areas. As presented here, the most important of these are: (1) it cannot calculate solutions at non-analytical points, such discontinuities and singularities; (2) calculation speeds are not competitive to those achieved by many other techniques; and (3) additional research is required to provide a theoretical basis for expansion point propagation and to develop tools for determining the radius of convergence for some calculations. Probably, most of these shortcomings can be eliminated. For example, it may be possible to create a similar procedure, using Laurent series coefficients, for situations where the solution is singular at one or more expansion points. Similarly, computational speeds should significantly increase with the creation and use of specially built software, operating similar to a modern commercial spreadsheet, but not including any of the unnecessary overhead. We have some definite thoughts on how that might be implemented. Finally, research should be forthcoming and fruitful because of the significant and undeniable capabilities demonstrated in the procedure.

In that last regard, perhaps we can best summarize by alluding to the situation involving the first problem of the biharmonic equation. In solving that problem, Marinos [2] noted

that Stephenson's solution method [3] had several weak points. These included an increased number of unknowns, which led to increased memory requirements, complicated equations and, when iteration was used, slow convergence or sometimes no convergence at all. Marinos concluded that these disadvantages may be necessary to achieve the high accuracy that the method produced. Based on our research, however, summarized in Subsection 3.1, that conclusion is no longer valid. In retrospect it is interesting that we could achieve the high accuracy without the disadvantages in that simulation, and in the others, using Taylor series mathematics. As is well known, Taylor series are rarely used to solve differential or integral equations. But as shown, Taylor solutions are very capable. To exploit that capability we simply needed a method to quickly and easily create, store, and manipulate a sufficient number of Taylor components to solve the equations to the desired level of accuracy.

REFERENCES

1. E. Kreyszig, *Advanced Engineering Mathematics* (Wiley, New York/London/Sydney, 1968), pp. 159, 636.
2. A. T. Marinos, Numerical experiments on compact computational schemes for solving the first biharmonic problem in rectangles, *J. Comput. Phys.* **115**, 406 (1994).
3. J. W. Stephenson, Single cell discretizations of order two and four for biharmonic problems, *J. Comput. Phys.* **55**, 65 (1984).
4. E. Y. Deeba and S. A. Khuri, A decomposition method for solving the nonlinear Klein-Gordon equation, *J. Comput. Phys.* **124**, 442 (1996).
5. E. Rouy and A. Tourin, A viscosity solutions approach to shape-from-shading, *SIAM J. Numer. Anal.* **29**, 867 (1992).